J Braz Comput Soc (2012) 18:311–330
DOI 10.1007/s13173-012-0069-3

WTI

D-Confidence: an active learning strategy to reduce label disclosure complexity in the presence of imbalanced class distributions

Nuno Filipe Escudeiro · Alípio Mário Jorge

Received: 2 August 2011 / Accepted: 6 March 2012 / Published online: 30 March 2012
© The Brazilian Computer Society 2012

Abstract In some classification tasks, such as those related to the automatic building and maintenance of text corpora, it is expensive to obtain labeled instances to train a classifier. In such circumstances it is common to have massive corpora where a few instances are labeled (typically a minority) while others are not. Semi-supervised learning techniques try to leverage the intrinsic information in unlabeled instances to improve classification models. However, these techniques assume that the labeled instances cover all the classes to learn which might not be the case. Moreover, when in the presence of an imbalanced class distribution, getting labeled instances from minority classes might be very costly, requiring extensive labeling, if queries are randomly selected. Active learning allows asking an oracle to label new instances, which are selected by criteria, aiming to reduce the labeling effort. D-Confidence is an active learning approach that is effective when in presence of imbalanced training sets. In this paper we evaluate the performance of d-Confidence in comparison to its baseline criteria over tabular and text datasets. We provide empirical evidence that d-Confidence reduces label disclosure complexity—which we have defined as the number of queries required to identify instances from all classes to learn—when in the presence of imbalanced data.

N.F. Escudeiro (✉)
DEI-ISEP, Instituto Politécnico do Porto, Porto, Portugal
e-mail: nfe@isep.ipp.pt

N.F. Escudeiro · A.M. Jorge
Laboratory of Artificial Intelligence and Decision Support,
INESC, Porto L.A., Portugal

A.M. Jorge
DCC-FCUP, Universidade do Porto, Porto, Portugal
e-mail: amjorge@fc.up.pt

Keywords Active learning · Imbalanced data · Label disclosure complexity · Text classification

1 Introduction

Classification tasks require a number of previously labeled instances. A major bottleneck is that instance labeling is a laborious task requiring significant human effort. This effort is particularly high in the case of text corpora and other unstructured data.

The effort required to retrieve representative labeled instances to learn a classification model is not only related to the number of distinct classes [2]. It is also related to class distribution in the available pool of instances. On a highly imbalanced class distribution, it is particularly demanding to identify instances from minority classes. These, however, may be important in terms of representativeness. Minority classes may correspond to specific information needs which are relevant for specific groups of users. In many situations, such as fraud detection, clinical diagnosis, news [35] and Web resource categorization [17], we face the problem of imbalanced class distributions.

The work described in this paper supports a broader goal related to the identification of representative instances for each class in the absence of previous descriptions of some or all the classes, in order to get a classification model that is able to fully recognize the target concept, including all the classes to learn no matter how frequent or rare they are. Furthermore, this must be achieved with a reduced number of labeled instances in order to reduce the labeling effort.

The aim of our current work is to evaluate the performance of our proposal, a new active learning strategy, w.r.t. its ability in finding representative instances of the classes to learn regardless of their distribution in the working set.

There are several learning schemes available for classification. The supervised setting allows users to specify arbitrary concepts. However, it requires a fully labeled training set, which is prohibitive when the labeling cost is high and, besides that, it requires labeled instances from all classes. Semi-supervised learning [11] allows users to state specific needs without requiring extensive labeling [17] but still requires that labeled instances fully cover the target concept. Unsupervised learning does not require any labeling but users have no chance to tailor clusters to their specific needs. Therefore there is no guarantee that the induced clusters are aligned with the classes to learn. In active learning, which seems more adequate to our goals, the learner is allowed to ask an oracle (typically a human) to label instances—these requests are called *queries*. The most informative queries are selected by the learning algorithm instead of being randomly selected as in supervised learning.

In this paper we describe and evaluate the performance of *d-Confidence* [19]. D-Confidence is an active learning approach that tends to explore unseen regions in instance space, thus selecting instances from unseen classes faster—with fewer queries—than traditional active learning approaches. D-Confidence selects queries based on a criterion that aggregates the posterior classifier confidence and the distance between queries and known classes. Confidence [4] and distance, farthest-first [23], are traditional active learning criteria. D-Confidence is biased towards instances that do not belong to known classes (low confidence) and that are located in unseen areas in instance space (high distance to known classes).

A workshop paper from 2008 [18] presents some preliminary results on the performance of d-Confidence. These results are based mainly on artificial datasets with the purpose of realizing the ability of d-Confidence in the early identification of rare instances.

These preliminary results were extended in [19]. This paper describes a systematic approach to the evaluation of d-Confidence. It is based on artificial data and focused on comparing the performance of d-Confidence to that of confidence w.r.t. the coverage of the instance space. Two-dimensional artificial datasets have been generated to exhibit a set of properties describing global dataset characteristics: cluster alignment, label distribution, cluster morphism and cluster separability. All these properties were defined as binary. Sixteen artificial datasets have been generated covering all the combinations of these four binary meta-descriptors expecting to simulate a wide range of real datasets' structures arising in classification tasks. The empirical results showed that d-Confidence selects queries from remote regions—where the density of known (labeled) instances is sparse—more efficiently than confidence. Instance space is covered more efficiently when using d-Confidence, thus creating conditions to identify representative cases from unknown classes earlier. On average, a 100 % coverage of the instance space is achieved by d-Confidence with a fraction of the effort required by confidence. Regarding the global properties of the datasets, d-Confidence performed clearly better than confidence on "well behaved" datasets (balanced, collinear, isomorphic and separable). On not so well behaved datasets, d-Confidence also performs better than confidence but not as clearly, especially with respect to the classification error.

D-Confidence, using SVM as the base classifier, was evaluated over text corpora in two workshop papers. In [20] we compare the performance of d-Confidence to that of confidence and random sampling, as a ground benchmark. The results from this paper show that d-Confidence identifies exemplary instances for all classes faster that confidence. This gain in labeling effort is bigger for minority classes, which are the ones where the benefits are more relevant for our purposes. As a consequence the classification model generated by d-Confidence is able of identifying more distinct classes faster. In [21] this work is continued by comparing d-Confidence performance on text corpora to its baseline criteria (confidence and farthest-first) with SVM base classifiers.

The current work extends previous results on d-Confidence providing a comprehensive description and evaluation of this active learning strategy. It adds several contributions, including a formal description of d-Confidence, the clear definition of its evaluation criteria, a comparative study of d-Confidence w.r.t. to different base classifiers, a systematic evaluation of the d-Confidence strategy against its baseline criteria over tabular and textual data with a main concern in the identification of rare instances in imbalanced data.

Our hypothesis is that d-Confidence improves the performance of both its baseline criteria. On the one hand, it improves the exploitation behavior of confidence, which is required to prevent excessive accuracy decrease; on the other hand, it improves the exploratory behavior of farthest-first, which is required to reduce the minimum number of queries needed to identify instances from all classes to learn.

Experimental outcomes led us to conclude that d-Confidence is more effective than confidence and farthest-first alone in achieving an homogeneous coverage of target classes.

In the rest of this paper we start by reviewing active learning, in Sect. 2. Section 3 describes d-Confidence. The evaluation process is presented in Sect. 4 and we state our conclusions and expectations for future work in Sect. 5.

2 Active learning

Active learning [4, 13, 33, 36] is a particular form of supervised learning where instances to label are selected by

the learner through some criteria aimed at reducing the label complexity [22], i.e., the number of label requests that are necessary and sufficient to learn the target concept.

In active learning, the learner is allowed to ask an oracle (typically a human) to label instances—these requests are called *queries*. The most informative queries, given the goals of the classification task, are selected by the learning algorithm instead of being randomly selected as is the case in passive supervised learning.

The term active learning has been originally coined in the education field in 1991, as a corollary of the broad discussion about instructional paradigms, which took place in the 1980s. It refers to the instructional activities involving students in doing things and thinking about what they are doing [8].

A few years before, the paradigm had already been applied to machine learning [4]. In this work the author sets a formal framework to study several types of query and their value for machine learning tasks. Although with some previous work performed by researchers, the term *active learning* seems to have been explicitly used in machine learning from 1994 on [13]. In this work the authors define active learning as any form of learning where the learner has some control over the input on which it trains.

Active learning approaches [13, 33, 36] reduce label complexity by analyzing unlabeled instances and selecting the most useful ones once labeled. Queries may be artificially generated [6]—the *query construction* paradigm—or selected from a pool [12] or a stream of data—the *query filtering* paradigm. Our current work is developed under the query filtering approach.

The general idea in active learning is to estimate the value of labeling one unlabeled instance. Query-By-Committee [38], for example, uses a set of classifiers to identify the instance with the highest disagreement. Schohn et al. [37] worked on active learning for Support Vector Machines (SVM) selecting queries—instances to be labeled—by their proximity to the dividing hyperplane. Their results are, in some cases, better than if all available data are used to train. Cohn et al. [14] describe an optimal solution for pool-based active learning that selects the instance that, once labeled and added to the training set, produces the minimum expected error. This approach, however, requires high computational effort. Previous active learning approaches (providing non-optimal solutions) aim at reducing uncertainty by selecting queries as the unlabeled instances on which the classifier is less confident [29].

Batch mode active learning—selecting a batch of queries instead of a single one before retraining—is useful when computational time for training is critical. Brinker [9] proposes a selection strategy, tailored for SVM, that combines closeness to the dividing hyperplane—ensuring a reduction in the version space [32] close to one half—with diversity among selected instances—ensuring that newly added instances provide additional reduction of version space. Hoi et al. [24] suggest a batch mode active learning relying on the Fisher information matrix to ensure small redundancy among selected instances. Li et al. [30] compute diversity within selected instances from their conditional error. Hoi et al. [25] use batch mode active learning to increase the number of labeled instances and its diversity to improve SVM performance in each iteration.

Dasgupta [15] defines theoretical bounds showing that active learning has exponentially smaller label complexity than supervised learning under some particular and restrictive constraints. Kääriäinen extended this work by relaxing some of those constraints [28]. An important conclusion of this work is that the gains of active learning are much more evident in the initial phase of the learning process, after which these gains degrade and the speed of learning drops to that of passive learning. Agnostic Active learning [5], A^2, achieves an exponential improvement over the usual label complexity of supervised learning in the presence of arbitrary forms of noise. This model is studied by Hanneke [22] setting general bounds on label complexity.

All these approaches assume that we have an initial labeled set covering all the classes of interest. However, this assumption does not necessarily hold. In fact, collecting and annotating cases is a critical—being one of the first stages it might limit the performance of the following—and demanding stage—requires domain specialists to retrieve and label exemplary instances for all target classes—in classification tasks [30]. The effort in finding these exemplary instances depends not only to the number of target classes [2] but also to their distribution in the working set. On a highly imbalanced class distribution, it is particularly demanding to identify examples from minority classes. These, however, may be important in terms of representativeness. This is the case of a document collection on the Web.

Clustering has also been explored to provide an initial structure to data or to suggest valuable queries. Tat et al. [34] incorporate clustering into active learning by learning a classification model from the set of the cluster representatives, and then propagates the classification decision to the other instances via a local noise model. The proposed model allows to select the most representative instances as well as to avoid repeatedly labeling instances in the same cluster. Adami et al. [2] merge clustering and oracle labeling to bootstrap a predefined hierarchy of classes. Although the original clusters provide some structure to the input, this approach still demands for a high validation effort, especially when these clusters are not aligned with class labels.

Huang et al. [27] explore the Wikipedia as a background knowledge base to create a concept-based representation of a text document enabling the automatic grouping of documents with similar themes. The semantic relatedness between Wikipedia concepts is used to find constraints for supervised clustering using active learning.

Dasgupta et al. [16] propose a cluster-based method that consistently improves label complexity over supervised learning. Their method detects and exploits clusters that are loosely aligned with class labels. The method has been applied to the detection of rare categories. It obtained significant gains in the number of queries that are required to discover at least one instance from each class. This latter work is in line with our own efforts for devising a method capable to swiftly identify instances from unknown classes. Preliminary results have been published by us also in 2008 in a workshop paper [18]. Hu et al. [26] propose an active learning schema, based on graph-theoretic clustering algorithms, to suppress the lack of ability from common active learning approaches in selecting new instances that belong to new classes that have not yet appeared in the working set, and the lack of adaptability to changes in the semantic interpretation of sample classes.

An important issue in active learning is the establishment of a compromise between *exploration*—finding representative instances in the dataset that are useful to label, focusing on completeness—and *exploitation*—sharpening the classification boundaries, focusing on accuracy.

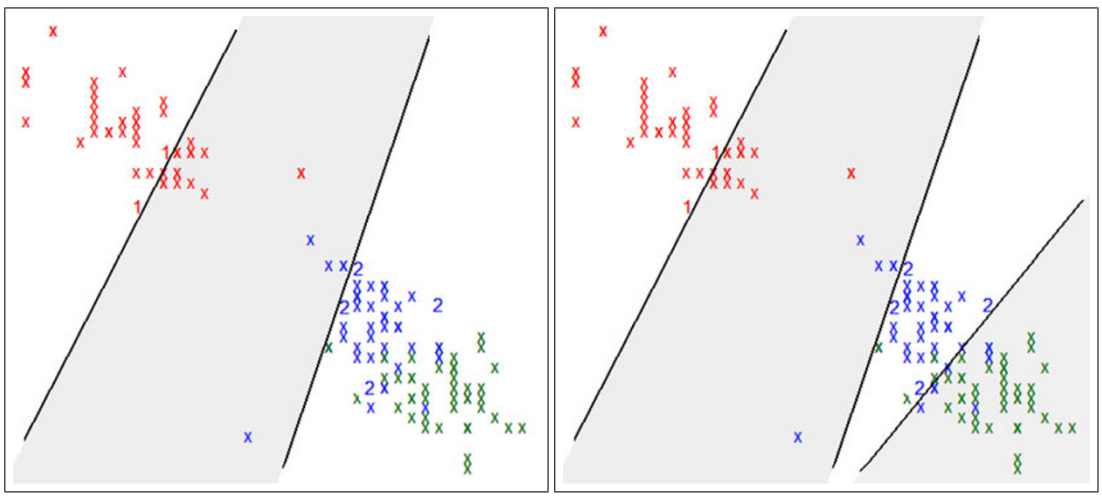
As described, common active learning methods select the queries which are closest to the decision boundary of the current classifier. They focus on improving the decision functions for previously labeled classes, i.e., they focus on exploitation. The work presented in this paper diverts classifier attention to other regions increasing the chances of finding new labels. D-Confidence adds an exploration bias to active learning.

3 D-Confidence active learning

Given a target concept with an arbitrary number of classes together with a sample of unlabeled examples from the target space (the working set), our purpose is to identify representative instances covering all classes while posing as few queries as possible, where a query consists of requesting a label to a specific instance. The working set is assumed to be representative of the class space—the representativeness assumption [31].

Active learners commonly search for queries in the neighborhood of the decision boundary (Fig. 1a), where class uncertainty is higher. The (perceived) uncertainty region is defined [13] as the area that is not determined by available information, i.e., the set of instances in the working set such that there are two hypotheses that are consistent with all training instances yet disagree on the classification of these instances. However, the perceived uncertainty region might be poorly mapping the real target concept, given current evidence.

Limiting instance selection to the perceived uncertainty region seems adequate when we have at least one labeled instance from each class in which case the perceived uncertainty region is probably consistent with the target concept. This class representativeness is assumed by the majority of active learning approaches. In such a scenario, selecting queries from the uncertainty region is very effective in reducing version space.



(a) Perceived uncertainty region (b) Real uncertainty region

Fig. 1 Uncertainty region (*shaded*). *n* represents labeled instances from class *n* and x represents unlabeled instances. We assume that the concept to learn has three distinct classes, one of which has not yet been identified

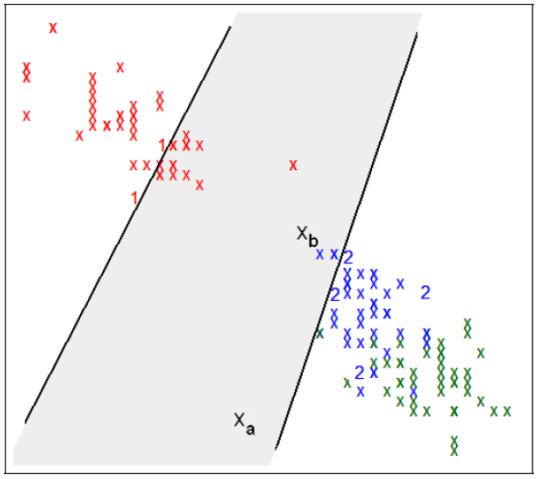


Fig. 2 For equally confident instances prefer those that are far from previously explored regions in instance space

But what if the real uncertainty region is not correctly or fully perceived by the current hypothesis? Under such an assumption, favoring exploitation rather than exploration withholds the chances to achieve an early complete coverage of the target concept.

3.1 The intuition

Our main concern is related to the initial phase of the learning process—data collection and annotation—when we are still looking for exemplary instances to characterize the concept to learn. Under these circumstances and while we do not have labeled instances covering all classes, the uncertainty region perceived by the active learner (Fig. 1a) is reduced to a portion of the real uncertainty region (Fig. 1b). Being limited to this partial view of the concept, the learner is more likely to waste queries. The amount of the uncertainty region that the learner misses is related to the number of classes in the concept to learn that have not yet been identified.

Our intuition (Fig. 2) is that query selection should be based not only on classifier confidence but also on distance to previously labeled instances. In the presence of two instances with equally low confidence—say, X_a and X_b in Fig. 2—we prefer to select the one that is farther apart from what we already know, i.e., from previously labeled instances—referring to Fig. 2 we would prefer to query X_a than X_b. This bias improves the exploratory behavior of the active learning approach.

3.2 D-Confidence

The most common active learning approaches rely on classifier confidence to select queries [4] and assume that the pre-labeled set covers all the labels to learn. The performance of these approaches is focused on accuracy, favoring exploitation over exploration. Our scenario is somehow different: we do not assume that we have pre-labeled instances from all classes and, besides accuracy, we are mainly concerned with the fast identification of representative instances from all classes.

To achieve our goals we propose a new selection criterion, *d-Confidence*, which deals well with under-represented classes. Instead of relying exclusively on classifier confidence we propose to select queries based on the ratio between classifier confidence and the distance to known classes. D-Confidence, weighs the confidence of the classifier with the inverse of the distance between the instance at hand and previously known classes.

D-Confidence is expected to favor a faster coverage of instance space, exhibiting a tendency to explore unknown regions. As a consequence, it provides better exploratory behavior than confidence alone. This drift towards unexplored regions and unknown classes is achieved by selecting the instance with the lowest d-Confidence as the next query. Lowest d-Confidence combines low confidence—probably indicating instances from unknown classes—with high distance to known classes—pointing to unseen regions in instance space. This effect produces significant differences in the behavior of the learning process. Common active learners focus on the uncertainty region, asking queries that are expected to narrow it down. The issue is that the portion of the uncertainty region that is perceived at a given moment is determined by the labels known at that moment. Focusing our search for queries exclusively in this region, while we are still looking for exemplary instances on some labels that are not yet known, is not effective. Unknown classes hardly come by unless they are represented in the current uncertainty region.

Algorithm 1 presents d-Confidence, an active learning proposal specially tailored to achieve a fast class representative coverage.

W is the working set, a representative sample of instances from the problem space. L_i is a subset of W. Members of L_i are the instances in W whose labels are known at iteration i. C_i is the set of the class labels that have representative instances in L_i. U, a subset of W, is the set of unlabeled instances present in the working set. At iteration i, U_i is the (set) difference between W and L_i; h_i represents the classifier learned at iteration i; q_i is the query selected at iteration i; $conf_i(u_j, c_k)$ is the posterior confidence on class c_k given instance u_j, at iteration i.

The core of our proposal is the computation of d-Confidence values for unlabeled instances; this is accomplished at the outer *for* cycle in Algorithm 1 as explained next. At step (11) we select the next query as the instance with the minimum d-Confidence. This query is then added to the labeled set (12) and the whole process iterates until

Algorithm 1 D-Confidence algorithm

(1) given W, L_1
(2) compute distance among instances in W
(3) $i = 1$
while not stopping criteria **do**
  (4) $U_i = W - L_i$
  (5) $C_i =$ distinct class labels in L_i
  (6) learn h_i from L_i
  (7) apply h_i to U_i generating $conf_i(u_j, c_k)$
  for $(u_j \in U_i)$ **do**
    for $(c_k \in C_i)$ **do**
      (8) $dist_i(u_j, c_k) = ClassDist(u_j, c_k)$
      (9) $dconf_i(u_j, c_k) = \frac{conf_i(u_j, c_k)}{dist_i(u_j, c_k)}$
    end for
    (10) $dConf_i(u_j) = \max_{c_k}(dconf_i(u_j, c_k))$
  end for
  (11) $q_i = \underset{u_j}{\operatorname{argmin}}(dConf_i(u_j))$
  (12) $L_{i+1} = L_i \cup \langle q_i, label(q_i)\rangle$
  (13) $i++$
end while

a given stopping criteria is met. At the current implementation, the learning process stops when the unlabeled pool is exhausted.

3.2.1 Computing d-Confidence

D-Confidence is obtained as the ratio between confidence and distance among unlabeled instances and known classes (1). We may view d-Confidence as the confidence per unit distance.

$$dConf(u) = \max_k \left(\frac{conf(u_j, c_k)}{dist(u_j, c_k)} \right) \tag{1}$$

For a given unlabeled instance, u_j, the classifier generates the posterior confidence w.r.t. known classes (7). The distance between unlabeled instance u_j and all labeled instances in class c_k, $dist()$, is computed by $ClassDist()$ at step (8). $ClassDist()$ is an indicator of the distance between one instance and one group of instances (those belonging to a given class). The Euclidean metric was previously used in step (2) to compute the distance between all pairs of instances in W. This distance indicator, $dist()$, is the median of the distances between instance u_j and all instances in class c_k. We expect the median to soften the effect of outliers. At step (9) we compute $dconf_i(u_j, c_k)$—the marginal d-Confidence for each known class, c_k, given the instance u_j—by dividing class confidence for a given instance by the aggregated distance to that class.

The maximum d-Confidence on individual classes for a given instance u_j is finally computed, at step (10), as the d-Confidence of the instance, $dConf_i(u_j)$.

3.2.2 Baseline criteria

D-Confidence aggregates two baseline criteria, confidence and distance (based on farthest-first). Confidence, generated at each iteration by the current version of the base classifier in use, is the posterior probability of class c_k given u_j. The aggregated distance to known classes, $dist_i(u_j, c_k)$, is computed by $ClassDist(u_j, c_k)$ based on the individual distances between each pair of instances (2). Individual pair distances might be computed by any distance function—at the current implementation we are using the Euclidean distance. $ClassDist(u_j, c_k)$ may also be any aggregation function computed on the individual pair distances between one unlabeled instance $u_j \in U_i$ and every labeled instance from class $c_K \in C_i$ known at iteration i—at the current implementation we are using the median.

$$ClassDist_i(u_j, c_k) = median\left(dist\left(u_j, C_i^k\right)\right) \tag{2}$$

C_i^k is the set of labeled instances known at iteration i that belong to class c_k, i.e., $C_i^k = \{\langle x, y\rangle \in L_i : y = c_k\}$.

3.3 Effect of d-Confidence on SVM

The output of SVM classifiers is the signed distance to the decision boundary measured in terms of half margin width—a case located on the decision boundary output 0 while an instance which is collinear with support vectors for class $+1$ generates an output 1 and an instance which is collinear with support vectors for class -1 generates an output -1. An instance with a distance to the decision boundary that is n times the distance between the boundary and a support vector output n. This distance, d, is transformed into $p \in [0, 1]$—representing the posterior confidence of the learner on class $+1$.

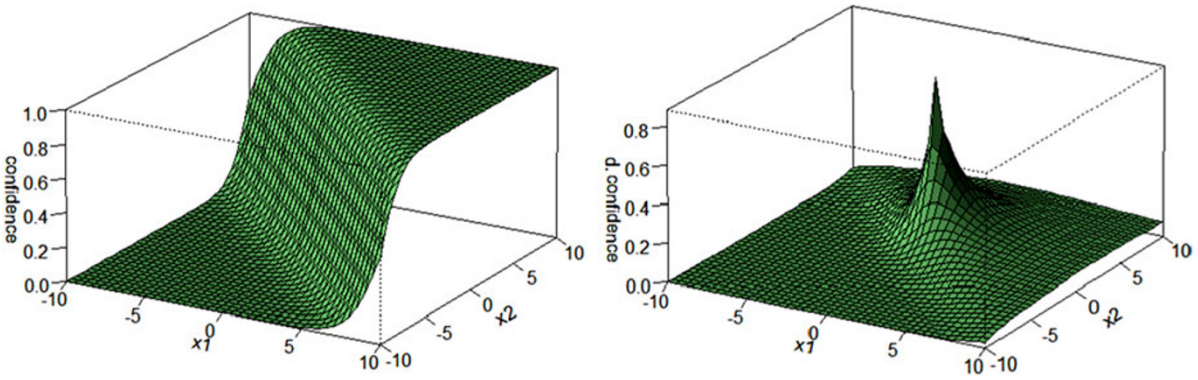
If, as is commonly the case, this transformation is based on logistic regression (3), the SVM classifier will be very confident on instances that lie far from the decision boundary (Fig. 3a), reducing the chances to select queries far from the current uncertainty region.

$$p = f(d) = \frac{1}{1 + e^{-d}} \tag{3}$$

To prevent this behavior and to direct the learner to low confidence instances but also to unexplored regions in instance space, the d-Confidence value of a point is high in the neighborhood of known instances decreasing with the distance to those (Fig. 3b).

4 Evaluating d-Confidence performance

The ultimate goal of our evaluation of d-Confidence is to assess its ability to identify instances from unseen classes



(a) Confidence for class +1 (b) D-Confidence for class +1

Fig. 3 Effect of d-Confidence for class +1 with an SVM classifier. We assume we have labeled instances near the point $(0, 0)$ of the input space (x_1, x_2). The decision boundary is the diagonal line from $(-10, 10)$ to $(10, -10)$

while querying for fewer labels without degrading accuracy when compared to its baseline criteria—confidence and farthest-first. We have designed our evaluation plan with several objectives in mind:

- first of all, we want to (a) compare the performance of d-Confidence against its baseline criteria;
- then we want to (b) assess the impact of the base classifier on the performance of d-Confidence;
- finally, we want to (c) determine whether the performance of d-Confidence depends on the dimensionality of the input feature space. In particular, we want to determine whether d-Confidence is appropriate for high-dimensional unstructured datasets, mainly text.

The evaluation was performed over several base classifiers, several datasets and several query selection criteria, including d-Confidence and its baseline criteria.

These objectives will be assessed from several performance indicators:

- *error* and *known classes* (see Definition 1), evaluated at each iteration throughout the learning cycle and
- *first-hit* (see Definition 2) and *label disclosure complexity* (see Definition 3), evaluated once for every combination of dataset, base classifier and query selection criterion.

4.1 Performance indicators

Our evaluation will be based on the performance indicators referred above: error, known classes, first-hit and label disclosure complexity.

To make these performance indicators clear, lets assume a generic classification task. C is the set of class labels to learn. $C_i \subseteq C$ is the set of class labels contained in a training set L_i.

Active learning is an iterative process requiring some prior initialization. C_1 is the set of labels that are represented in L_1, the initialization training set. At each iteration, new labeled instances, called queries, are added to the training set.

Error, is a common assessment criterion for classification tasks. We have computed the progress of the generalization error—the error in the test set—over all iterations as new labeled instances are added to the training set.

Known classes is the number of classes that have representative labeled instances in the training set at a given iteration.

Definition 1 Known classes, kc_i is the cardinality of C_i, i.e., the number of classes given for learning.

First-hit is defined for each class. It is the number of queries required to identify the first instance of the class for a given dataset, base classifier and query selection criterion.

Definition 2 For each $c_k \in C$, $c_k \notin C_1$, first-hit, fh_k, is the number of queries required to identify the first instance of class c_k. The initialization queries, the instances in L_1, are not accounted for.

Label disclosure complexity (LDC) aims to evaluate the ability of the learning process to reveal all the classes belonging to the concept to learn. LDC is inspired on label complexity [22], defined under the active learning setting as the number of queries that are sufficient and necessary to learn the target concept. LDC is the minimum number of queries being required to identify at least one instance from every class to learn. LDC equals the maximum first-hit computed over all the classes for a given combination of dataset, base classifier and query selection criterion.

Definition 3 Label disclosure complexity (LDC) is the minimum number of queries that are required to identify at

Table 1 Class distribution in tabular datasets

Dataset	#Instances	#Features	1	2	3	4	5	6	7	8	9	10	11
Iris	150	4	50	50	50								
Cleveland	298	13	161	53	36	35	13						
Vowels	330	10	30	30	30	30	30	30	30	30	30	30	30
Satlog	500	36	125	118	96	67	48	46					
Poker	500	10	270	170	34	12	4	3	3	2	1	1	

least one instance from every $c_k \in C$. LDC is equal to $\max_k(fh_k)$.

4.2 Experimental setting

The evaluation plan includes two phases, A and B.

Phase A covers objectives (a) and (b) set above (Sect. 4). The experiments in this phase were performed over tabular data. We have used five datasets from the UCI repository [1]:

- Iris (one class is separable while the other two are not),
- Cleveland heart disease (imbalanced class distribution),
- a random sample from Vowels (higher number of distinct classes than the others),
- a sample from Satlog (higher number of attributes than the others) and
- a sample from Poker (highly imbalanced class distribution).

These datasets were selected for their properties, mainly due to their distinct class distributions (Table 1).

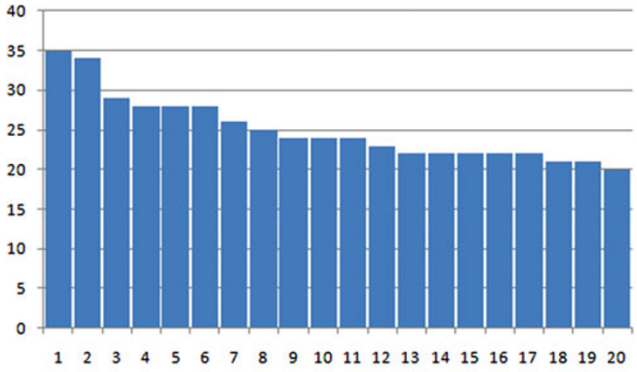
The purpose of phase A is to assess d-Confidence on regular data avoiding to add extra disturbing factors that might come by when using unstructured data. As base classifiers, we have used a neural network (NNET), a decision tree (RPART) and Support Vector Machine classifiers with linear kernels (SVM).

Phase B covers objective (c) set above (Sect. 4). For phase B we have selected two high-dimensional unstructured datasets. Two samples from traditional text corpora were used:

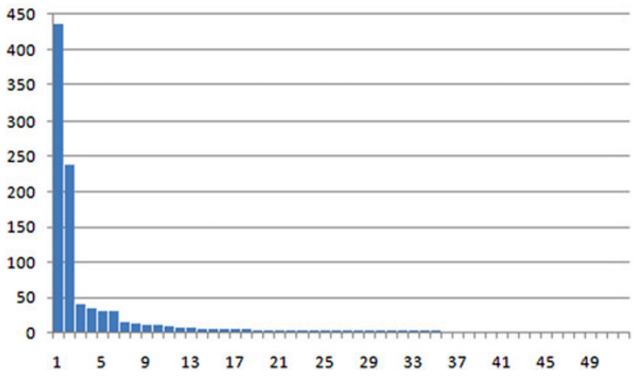
- a stratified sample from the 20 Newsgroups corpus (NG), containing 500 documents described by 10333 terms, and
- a stratified sample from the R52 set of the Reuters-21578 collection (R52), containing 1000 documents described by 6019 terms.

The NG dataset has documents from 20 distinct classes while the R52 dataset has documents from 52 distinct classes.

These datasets have been selected for their distinct class distributions. The class distribution in NG is fairly balanced (Fig. 4a) with a maximum frequency of 35 and a minimum frequency of 20 while the R52 dataset presents an highly imbalanced class distribution (Fig. 4b). The most frequent



(a) NG corpus



(b) R52 corpus

Fig. 4 Class distribution in text corpora

class in R52 has a frequency of 435 while the least frequent has only two instances in the dataset. This dataset has 42 classes, out of 52, with a frequency below 10 from which 31 are below 5.

In phase B we have used SVM classifiers in all experiments. SVM are commonly referred as being among the most accurate classifiers for high-dimensional input spaces, in general, and text, in particular [10].

The query selection criteria under evaluation are d-Confidence and its baseline criteria: standard confidence and farthest-first. The performance of these criteria on all datasets was estimated with 10-fold cross-validation. Folds are stratified random samples comprising a partition of the working set. Our aim is to compute the number of queries

that are required to identify at least one instance from all classes—from which we can compute known classes, first-hit and LDC—and to compute generalization error.

The labels in the training set are initially hidden from the classifier being revealed as the learning process iterates. For each iteration, the active learning algorithm asks for the label of a single instance. For the initialization of each fold we give two pre-labeled instances—from two distinct classes—to the classifier. These are randomly selected from the training set. Initialization instances for a given fold with labels already selected are disregarded. Given the fold, the same initial instances are used for all experiments.

The Poker dataset has a highly imbalanced dataset causing some exceptions. The two classes with frequency 1 from the Poker dataset are never selected as initial classes. Two out of the 10 folds used for cross-validation do not include all the 10 classes in the Poker dataset. For this reason, the maximum number of classes found when using this dataset is below the total number of classes in the dataset, since it is estimated as a mean over all folds.

In all the experiments, in both phases, we have compared our d-Confidence proposal against its baseline selection criteria: the common confidence active learning setting—where query selection is solely based on low posterior confidence of the current classifier—and farthest-first—where query selection is based only on distance from training instances which is independent from the base classifier. Comparing these criteria against each other provides evidence on the performance gains, or losses, of d-Confidence when compared to its baselines: confidence, and distance (farthest-first).

We have performed significance t-tests for the differences of the means observed when using farthest-first, confidence and d-Confidence. Statistically different means (significance level of 5 %) are presented in bold face.

In some cases we are using samples extracted from the whole dataset with fewer instances than those available. There is no loss of generality arising from this fact since the learning process converges, in respect to the indicators being measured, before those samples are exhausted.

4.3 Empirical results from phase A

In the first experimental phase we want to assess the ability of d-Confidence to reduce LDC over its baseline criteria. In parallel we evaluate accuracy as well. This assessment was performed over a set of base classifiers to evaluate their effect on the performance of d-Confidence.

In every experiment the training set starts with two pre-labeled instances. At each iteration a new instance is queried for its label and added to the training set.

We have recorded the number of distinct labels identified and the error on the test set for each iteration, for every combination of dataset, base classifier and query selection criteria. From these, we have then computed the mean number of known classes and mean generalization error in each iteration over all cross-validation folds.

The evolution of the error rate and the number of known classes for each dataset, when using SVM as a base classifier, is shown in Figs. 5a–5e with curves for each selection criteria under evaluation.[1] For convenience of representation, the mean number of known classes has been normalized to the total number of classes in the dataset thus being transformed into the percentage of known classes instead of the absolute number of known classes. This way the number of known classes and generalization error are both bounded in the same range (between 0 and 1) and we can conveniently represent them on the same chart. Means at each iteration are micro-averages—all the instances are equally weighted—over all cross-validation folds for a given combination of dataset, classifier and selection criterion.

The evolution of these indicators—generalization error and mean number of known classes—throughout all the learning cycle can be summed up to provide evidence on overall performance. Means in Table 2 are micro-averages over all iterations for a given combination of dataset, classifier and query selection criteria, providing a perspective of the average performance of the query strategy throughout the learning cycle.

Besides the overall error and number of known classes we have also observed first-hit (Table 3). When computing first-hit for a given class we have omitted the experiments where the labeled set for the first iteration contains that class, following Definition 2.

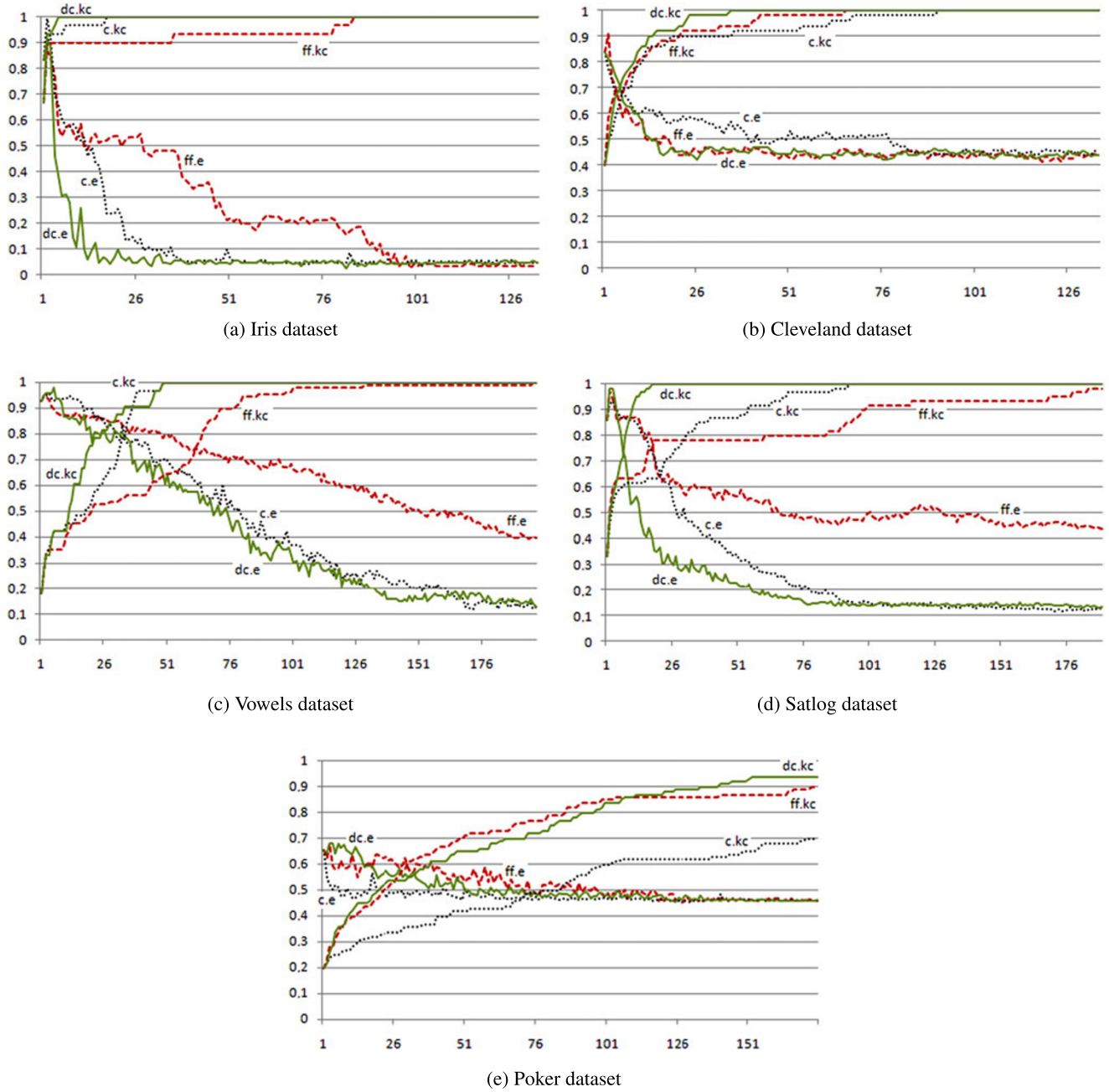
From first-hit we compute LDC for each scenario (Table 4). LDC is the maximum first-hit for a given scenario. It provides the number of queries that are required by the active learning strategy to identify at least one instance from each class to learn, i.e., to achieve full coverage of the target concept.

4.4 Analysis of results from phase A

In phase A we evaluate the performance of d-Confidence over tabular data w.r.t. representativeness, accuracy and first-hit. The influence of the base classifier on the learning strategy is also evaluated.

If we focus on SVM, which will be our base classifier for text corpora, we can observe in Table 2 that d-Confidence performs better than confidence and farthest-first, both at labeling effort and accuracy, over tabular datasets. The only

[1] We will use the following notation to refer to results in tables and charts: *ff* stands for farthest-first, *c* stands for confidence and *dc* stands for d-Confidence. Generalization error will be referred by *e*, *kc* will refer to the mean number of known classes and *ldc* refers to LDC.



(a) Iris dataset

(b) Cleveland dataset

(c) Vowels dataset

(d) Satlog dataset

(e) Poker dataset

Fig. 5 Known classes and generalization error in tabular data (when using SVM as the base classifier)

exception occurs at the Poker dataset where the mean error over all the learning process is lower when using confidence.

The dominance of d-Confidence throughout all the learning process is also observable from Fig. 5. This dominance is clear, both in terms of error and known classes, at Iris, Vowels and Satlog (Figs. 5a, 5c and 5d). Iris and Vowels have uniform class distributions while Satlog has a fairly balanced class distribution with a coefficient of variation equal to 42 %—the coefficient of variation is the ratio of the standard deviation to the mean. The same performance is also evident at the Cleveland dataset (Fig. 5b). Here, however, while the gain of d-Confidence over confidence is clear it is not as salient over farthest-first. The Cleveland dataset has one majority class with a frequency over 50 % and one under-represented class with frequency below 5 %. The coefficient of variation is equal to 98 %. At the highly imbalanced Poker dataset (Table 1) d-Confidence takes clear advantage over confidence w.r.t. known classes over all the learning process (Fig. 5e). We can also observe that d-Confidence is outperformed by farthest-first w.r.t. known

Table 2 Micro-averaged number of known classes and error. Means have been computed over all iterations from all cross-validation folds for every combination of dataset, classifier and query selection criteria

Dataset	Classifier	ff.kc	c.kc	dc.kc	ff.e	c.e	dc.e
Iris	SVM	2.8	3.0	3.0	0.257	0.134	**0.082**
Iris	NNET	2.8	2.7	**3.0**	0.14	0.164	**0.05**
Iris	RPART	2.8	3.0	3.0	0.342	0.187	0.184
Cleveland	SVM	4.9	4.8	4.9	0.451	0.473	0.45
Cleveland	NNET	4.9	4.9	4.9	0.464	0.465	**0.447**
Cleveland	RPART	4.9	4.9	4.9	0.479	0.496	0.485
Poker	SVM	8.7	7.2	**8.8**	0.484	0.466	0.484
Poker	NNET	8.7	7.8	**8.8**	0.526	0.49	0.49
Poker	RPART	8.7	7.5	8.6	0.524	**0.495**	0.517
Satlog	SVM	5.6	5.8	**6**0	0.349	0.186	**0.162**
Satlog	NNET	5.6	5.9	5.9	0.729	0.726	0.739
Satlog	RPART	5.6	5.9	6.0	0.430	**0.261**	0.28
Vowels	SVM	9.8	10.4	**10.5**	0.546	0.341	**0.322**
Vowels	NNET	9.8	10.7	10.6	0.661	0.601	0.623
Vowels	RPART	9.8	10.7	10.5	0.645	0.617	0.632

classes at the initial quarter of the learning process—up to iteration 106—but overcomes it from there on. At this dataset, however, the error of d-Confidence is clearly dominated by that of confidence at the initial stage of the learning process.

The differences in mean error gains are statistically significant at the Iris, Satlog and Vowels datasets in favor of d-Confidence. At the other datasets—Cleveland and Poker—the difference is not statistically significant. The most relevant evidence is probably the fact that error does not degrade; in fact, it generally improves when using d-Confidence, when compared to confidence and farthest-first, with SVM base classifiers.

If we move now to the other classifiers—NNET (neural network) and RPART (decision tree)—over tabular data, we can observe a similar dominance. D-Confidence achieves higher or equal means of known classes on all combinations except when using NNET and RPART over the Vowels dataset and RPART over Poker (Table 2). When it comes to the mean error rate, d-Confidence does not perform has well has when relying on a SVM base classifier. D-Confidence presents a lower mean error at the Iris dataset, when using neural networks or decision trees, and also at Cleveland and Poker when using NNET. At the other combinations, the error observed when using d-Confidence as a query selection strategy is outperformed by the other strategies, although with no statistical significance.

D-Confidence also outperforms confidence first-hit performance, in general. The same does not hold when comparing d-Confidence and farthest-first w.r.t. first-hit where we do not perceive clear evidence on the best performer.

If we sum the number of classes over all datasets, we can find a total of 35 classes over the five tabular datasets (three from Iris, five from Cleveland, 10 from Poker, six from Satlog and 11 from Vowels). These datasets have been submitted to three distinct classifiers (SVM, NNET and RPART). In total, for all the experiments, we have evaluated 105 classes.

We can observe that confidence first-hits classes before d-Confidence only on 33 out of these 105 classes (Table 3). From these 33 cases, eight happen when using SVM as a base classifier, 12 when using NNET and 13 when using RPART. It is worthwhile noting that 17 out of these 33 cases occur at the Vowels dataset. The Vowels dataset has a uniform class distribution (30 instances per class). The added value of d-Confidence is more evident at imbalanced datasets.

The Poker dataset—where two out of ten classes occur only on a single case corresponding to a relative frequency of 0.2 % and six classes have a frequency below 1 %—allows evaluating the early identification of under-represented classes. The average first-hit computed from Table 3 over under-represented classes—classes 5 to 10—shows that confidence is not appropriate to find rare instances (Table 5).

D-Confidence outperforms both its baseline criteria w.r.t. the early identification of instances from under-represented classes when using SVM and NNET as base classifiers. Farthest-first however, takes the lead when using decision trees (RPART).

LDC provides further evidence supporting the improved performance of d-Confidence over its baseline criteria. In fact, d-Confidence has the lowest LDC on all combinations of dataset and classifier that have been evaluated on tabular data except on the Vowels dataset when using RPART as a base classifier (Table 4). The average gain on d-Confidence LDC for all pairs dataset/classifier, when compared to confidence, on tabular data is of 542 %, meaning that confidence

Table 3 Mean number of queries required to first hit unknown classes

Dataset	Classifier	A.L.	1	2	3	4	5	6	7	8	9	10	11
Iris	SVM	ff	1.0	65.3	1.0								
Iris	SVM	c	1.0	6.7	2.7								
Iris	SVM	dc	1.0	2.7	1.0								
Iris	NNET	ff	1.0	65.3	1.0								
Iris	NNET	c	37.5	1.0	83.0								
Iris	NNET	dc	1.0	1.3	1.0								
Iris	RPART	ff	1.0	65.3	1.0								
Iris	RPART	c	1.0	2.0	3.3								
Iris	RPART	dc	1.0	1.7	1.0								
Cleveland	SVM	ff	3.2	12.5	13.5	2.3	24.2						
Cleveland	SVM	c	2.5	7.0	8.3	19.0	39.8						
Cleveland	SVM	dc	2.7	14.5	8.3	4.8	8.0						
Cleveland	NNET	ff	3.2	12.5	13.5	2.3	24.2						
Cleveland	NNET	c	2.2	2.8	5.3	3.5	16.2						
Cleveland	NNET	dc	1.7	9.8	4.7	3.5	10.5						
Cleveland	RPART	ff	3.2	12.5	13.5	2.3	24.2						
Cleveland	RPART	c	3.0	1.0	17.7	4.3	16.2						
Cleveland	RPART	dc	2.2	13.2	3.5	4	5.3						
Poker	SVM	ff	4.5	2.0	2.9	17.2	27.6	85.1	39.1	63.5	200.4	63.7	
Poker	SVM	c	1.0	3.0	19.5	42.8	112.5	112.2	146.9	222.9	250.9	248.8	
Poker	SVM	dc	3.0	2.0	4.6	9.0	45.0	96.6	98.2	68.1	90.0	58.8	
Poker	NNET	ff	4.5	2.0	2.9	17.2	27.6	85.1	39.1	63.5	200.4	63.7	
Poker	NNET	c	2.5	1.0	12.5	41.2	74.7	145.3	177.5	67.0	70.6	311.6	
Poker	NNET	dc	2.0	2.0	7.0	26.1	49.2	38.7	74.0	63.6	114.2	95.1	
Poker	RPART	ff	4.5	2.0	2.9	17.2	27.6	85.1	39.1	63.5	200.4	63.7	
Poker	RPART	c	1.0	3.0	29.0	48.0	34.1	116.8	124.5	211.6	326.5	155.9	
Poker	RPART	dc	2.5	2.0	5.6	11.3	24.9	89.0	83.8	73.0	168.4	92.0	
Satlog	SVM	ff	68.0	107.0	20.9	1.6	1.1	95.8					
Satlog	SVM	c	11.5	5.2	34.1	31.6	28.1	23.1					
Satlog	SVM	dc	8.8	9.6	4.4	3.0	1.1	9.5					
Satlog	NNET	ff	68.0	107.0	20.9	1.6	1.1	95.8					
Satlog	NNET	c	4.8	6.6	7.9	2.5	24.4	6.2					
Satlog	NNET	dc	3.5	8.4	5.0	2.5	1.2	15.9					
Satlog	RPART	ff	68.0	107.0	20.9	1.6	1.1	95.8					
Satlog	RPART	c	5.8	1.0	2.3	10.9	16.0	7.8					
Satlog	RPART	dc	7.8	7.4	4.1	2.4	1.1	11.4					
Vowels	SVM	ff	1.1	13.0	22.6	52.2	60.5	71.4	66.4	8.2	62.1	3.9	88.6
Vowels	SVM	c	2.5	10.0	14.0	31.0	12.3	27.3	29.0	15.0	31.3	18.3	24.0
Vowels	SVM	dc	2.0	12.0	19.0	16.0	24.3	26.3	23.3	2.3	25.7	3.0	22.7
Vowels	NNET	ff	1.1	13.0	22.6	52.2	60.5	71.4	66.4	8.2	62.1	3.9	88.6
Vowels	NNET	c	27.3	13.0	4.5	7.1	13.8	7.5	9.5	14.8	5.6	11.8	5.9
Vowels	NNET	dc	3.6	8.4	15.9	7.8	21.6	15.5	9.5	7.5	11.9	4.8	24.3
Vowels	RPART	ff	1.1	13.0	22.6	52.2	60.5	71.4	66.4	8.2	62.1	3.9	88.6
Vowels	RPART	c	2.0	31.6	17.8	4.9	2.8	12.8	10.0	3.8	12.9	11.8	4.0
Vowels	RPART	dc	1.3	8.0	39.0	17.1	13.2	30.9	10.5	3.2	26.6	6.2	39.9

Table 4 LDC for tabular datasets

Dataset	Classifier	ff.ldc	c.ldc	dc.ldc	Best
Iris	SVM	65.3	6.7	2.7	dc
Iris	NNET	65.3	83.0	1.3	dc
Iris	RPART	65.3	3.3	1.7	dc
Cleveland	SVM	24.2	39.8	14.5	dc
Cleveland	NNET	24.2	16.2	10.5	dc
Cleveland	RPART	24.2	17.7	13.2	dc
Poker	SVM	200.4	250.9	98.2	dc
Poker	NNET	200.4	311.6	114.2	dc
Poker	RPART	200.4	326.5	168.4	dc
Satlog	SVM	107.0	34.1	9.6	dc
Satlog	NNET	107.0	24.4	15.9	dc
Satlog	RPART	107.0	16.0	11.4	dc
Vowels	SVM	88.6	31.3	26.3	dc
Vowels	NNET	88.6	27.3	24.3	dc
Vowels	RPART	88.6	31.6	39.9	c

Table 5 Average first-hit over under-represented classes at the Poker dataset

Classifier	ff	c	dc
SVM	80	182	76
NNET	80	141	72
RPART	80	162	89

requires over six times more queries than d-Confidence to identify all class labels. This figure, however, is highly biased by the outlier observed on Iris/NNET. Nevertheless, if we remove this outlier from our data we still have a gain of 101 % in LDC, meaning that, on average, confidence requires twice as many queries as d-Confidence to achieve a full coverage of the classes to learn on all tabular datasets.

4.4.1 Performance under different levels of class imbalance

With the purpose of reinforcing the previous evidence supporting the ability of d-Confidence when in presence of imbalanced data, we have evaluated the active learning strategies being studied under different levels of class imbalance. We have based this evaluation on the two datasets exhibiting a uniform distribution—Iris and Vowels—and on SVM base classifiers. The original training datasets were manipulated to ensure imbalanced class distributions. We have randomly sampled from each training fold a set of instances from given classes to be removed from the training dataset, thus achieving biased distributions with minority classes. Then we have repeated the learning process as before to these training data and collected the results described below.

From each dataset we have extracted four samples according to the process briefly described above. At Iris the number of instances from one of the classes—which will become the minority class—was reduced in those samples to 1, 3, 5 and 9, corresponding to a percentage of 2 %, 6 %, 11 % and 19 % relative to the frequency of each of the two remaining classes which kept their uniform distribution from the original training dataset.

Table 6 LDC under different imbalance levels. SVM as base classifier. Imbalance is the ratio of the frequency of the minority classes to the other classes

Dataset	Imbalance	ff.ldc	c.ldc	dc.ldc	Best
Iris	19 %	84	61	3	dc
Iris	11 %	87	61	3	dc
Iris	6 %	87	61	4	dc
Iris	2 %	88	88	5	dc
Vowels	21 %	99	26	23	dc
Vowels	10 %	84	39	35	dc
Vowels	7 %	98	69	55	dc
Vowels	3 %	102	58	74	c

At Vowels, a dataset with 11 classes, the number of instances from four of them—which will become the minority classes—was reduced in those samples to 1, 2, 3 and 6, corresponding to a percentage of 3 %, 7 %, 10 % and 21 % relative to the frequency of each of the remaining classes which kept their uniform distribution from the original training dataset.

The LDC computed from these experiments (Table 6) confirms the ability of d-Confidence to retrieve rare instances in comparison to its baseline criteria.

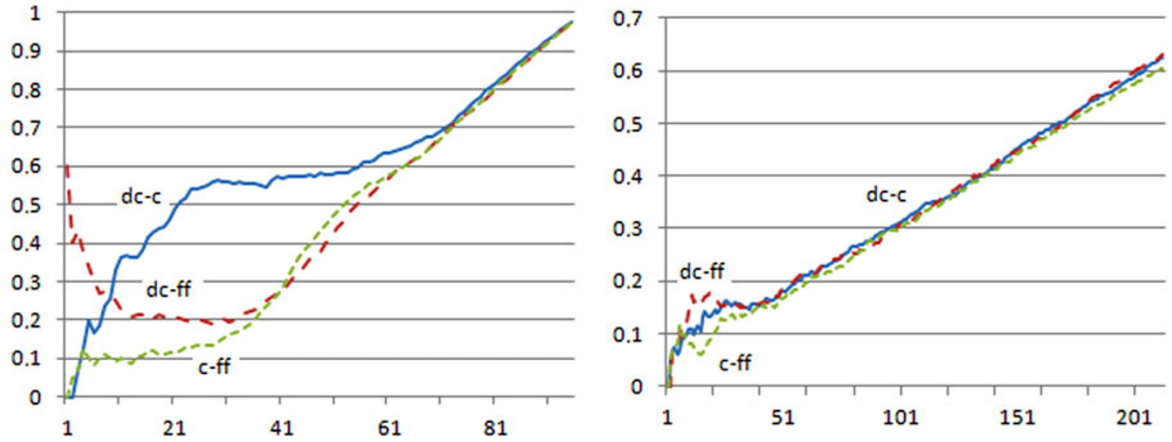
On average, d-Confidence presents a lower LDC than its baseline criteria on all settings except at Vowels with 3 % imbalance. We may observe the same scenario, with a significant dominance by d-Confidence, when analyzing the empirical results on the number of known classes and on error (Table 7). D-Confidence outperforms its baseline criteria with statistical significance at all settings except at the Vowels dataset with 21 % imbalance.

4.4.2 Common queries selection

Comparing the instances that are selected by each active learning strategy adds relevant information to our discussion. Are all strategies selecting the same instances at the same time throughout the learning cycle? We have investigated this question by measuring the percentage of common selected queries as the learning process iterates at Iris (Fig. 6a) and Vowels (Fig. 6b). Each curve in charts represents the average, computed over all cross-validation folds at each iteration, of the percentage of common instances observed in the labeled sets used to train the classifier under the referred strategies—d-Confidence (dc), confidence (c) or farthest-first (ff).

Table 7 Micro-averaged number of known classes and error. Means have been computed over all iterations from all cross-validation folds for every combination of dataset, imbalance level and query selection criteria. Bold faced values are statistically significant at 5 %

Dataset	Imbalance	ff.kc	c.kc	dc.kc	ff.e	c.e	dc.e
Iris	2 %	2.56	2.48	**2.96**	0.72	0.79	**0.67**
Iris	6 %	2.60	2.58	**2.98**	0.63	0.59	**0.40**
Iris	11 %	2.61	2.59	**2.98**	0.58	0.49	**0.26**
Iris	19 %	2.64	2.62	**2.98**	0.52	0.41	**0.15**
Vowels	3 %	8.11	8.87	**8.98**	0.94	0.92	**0.92**
Vowels	7 %	8.40	9.36	**9.55**	0.92	0.92	**0.91**
Vowels	10 %	8.65	9.77	**10.12**	0.91	0.89	**0.89**
Vowels	21 %	8.83	10.27	10.28	0.81	0.77	**0.76**



(a) Iris (imbalance 19%) (b) Vowels (imbalance 21%)

Fig. 6 Evolution of the percentage of common selected queries throughout the learning cycle. Each line represents the percentage of common instances for a given pair of strategies (dc-c, dc-ff, c-ff)

It is clear from Fig. 6a that d-Confidence and confidence query many common instances during the initial stage of the learning process at the Iris dataset. In fact, after the first 29 queries the labeled sets of both these strategies have nearly 60 % intersection. This level of overlapping then stabilizes to start increasing later as a consequence of the exhaustion of the unlabeled set which necessarily increases the interception between the labeled sets of all strategies.

The opposite behavior is observed when comparing farthest-first with either confidence or d-Confidence. Despite the fact that d-Confidence and farthest-first share many common instances at the very first iterations (60 %), this overlap drops fast getting close to 20 % after 11 queries.

At the Vowels dataset, the overlap between the labeled sets being built by all active learning strategies increases at a constant rate throughout the majority of the learning process. Only at the very beginning, during the initial 35 iterations, there is some difference in this behavior with d-Confidence and farthest-first querying more common instances than the rest. As observed also at the Iris dataset, confidence and farthest-first are the strategies sharing less queries.

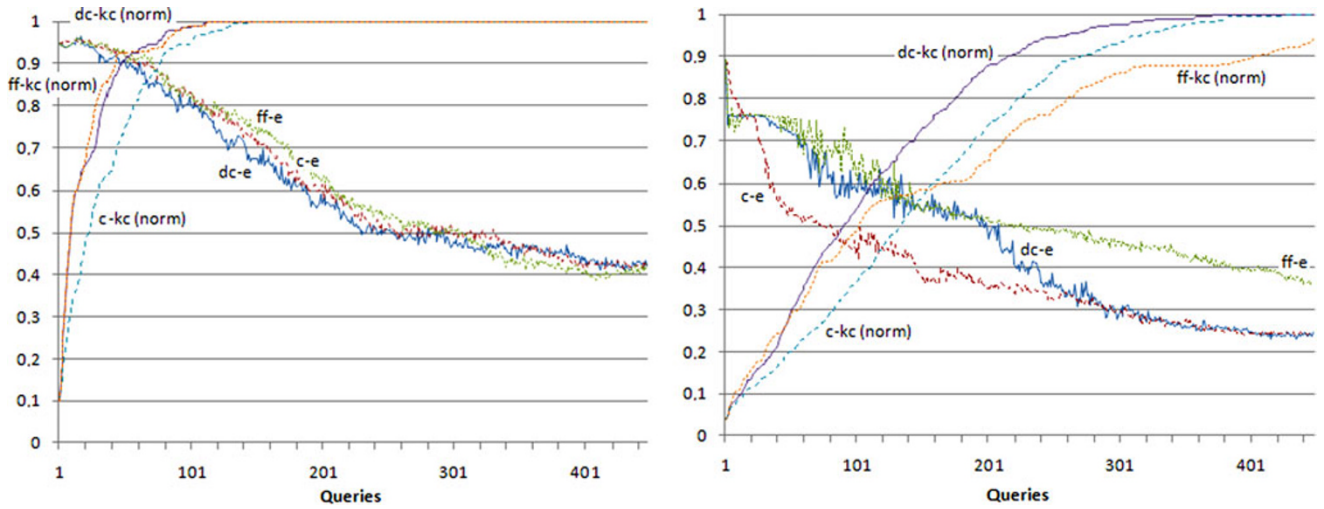
4.5 Empirical results from phase B

The evolution of the error rate and the number of known classes over text corpora is shown in Figs. 7a and 7b with curves for each selection strategy under evaluation.

Similarly to what we have done for phase A, the evolution of error and mean number of known classes throughout all the learning cycle has been also summed up to summarize overall performance on text corpora (Table 8).

Besides the overall number of queries required to retrieve labels from all classes and generalization error, we have also observed first-hit (Tables 9 and 10). When computing first-hit for a given class we have excluded the experiments where the labeled set for the first iteration contains instances from that class.

The learning process for the R52 dataset was halted after 600 iterations, before exploring the full unlabeled pool—the working set had 1000 instances, 900 of which were used for training in each fold. All the class labels to learn were identified after 600 iterations for all the selection criteria, except for farthest-first. The mean number of known classes after 600 iterations equals 52 for confidence and d-Confidence,



(a) NG corpus (b) R52 corpus

Fig. 7 Known classes and generalization error

Table 8 Micro-averaged number of known classes and error. Means have been computed over all iterations from all cross-validation folds for every combination of dataset, classifier and query selection criteria

Dataset	Classifier	ff.kc	c.kc	dc.kc	ff.e	c.e	dc.e
NG	SVM	19.2	18.6	19.1	0.631	0.629	**0.612**
R52	SVM	34.4	35.5	**39.7**	0.531	**0.383**	0.447

meaning these criteria have achieved full coverage of the class labels to learn in all the cross-validation folds. For farthest-first this mean is 50.3 which means that farthest first cannot identify all class labels in all cross-validation folds after 600 iterations. Farthest-first missed in several folds, six classes with frequency of two, two classes with frequencies of three and one class with a frequency of four. In such cases we have assigned a first-hit value of 601 for the unidentified classes. For instance, in a given fold where farthest-first misses two classes their first-hit values are assumed to be 601 and 602—the very first queries after halting the learning process at 600 iterations. First-hit means were computed on this assumption. Under such circumstances this is the most favorable assumption for farthest-first.

We have computed LDC—the number of queries that are required to identify at least one instance from each class to learn—from first-hit, according to Definition 3 for each scenario (Table 11).

4.6 Analysis of results from phase B

Figure 7a shows that there is no clear dominance, neither from d-Confidence nor from farthest-first, when finding unknown classes in the NG dataset. However, both these criteria outperform confidence at this dataset. The difference between mean first-hit of d-Confidence and farthest-first in Table 9—20.45 for farthest-first and 21.57 for d-Confidence—is not statistically significant, at a 5 % significance level.

D-Confidence accuracy dominates that of farthest-first (Fig. 7b). The mean accuracy of d-Confidence over all iterations is 2 % better that the one of farthest-first. This result is significant at 5 % significance. The NG dataset has a fairly balanced class distribution. On the R52 dataset, which has an highly imbalanced class distribution, we can observe very distinctive performance (Fig. 7b).

In R52, farthest-first starts by identifying unknown classes a little faster than d-Confidence (Fig. 7a). However, after the initial learning stage, d-Confidence outperforms and dominates farthest-first. When identifying unknown classes, farthest-first leads, up to the 45th query, on average, taking a maximum advantage of two classes after 37 queries. After 45 queries, with 13.2 classes identified on average, d-Confidence clearly dominates farthest-first.

It is interesting to notice that farthest-first beats d-Confidence on the majority classes (Tables 10) but, when all majority classes are found and only minority classes are left unexposed, d-Confidence reveals its ability to find rare instances. The mean frequency of the classes that are first found in R52 by d-Confidence is 3.2, while it is 12.5 for confidence and 33.8 for farthest-first.

If we take a step back to analyze d-Confidence first-hit against farthest-first on the highly imbalanced Poker

Table 9 First-hit for the NG dataset

Class	Freq	ff-fh	c-fh	dc-fh
1	29	**29.8**	36.9	35.7
2	22	45.4	46.6	45.7
3	21	87.9	**63.7**	85.4
4	34	7.5	29.4	7.4
5	35	**22.2**	23.6	25.2
6	24	17.6	41.2	17.1
7	21	11.4	59.6	12.6
8	24	12.6	32.9	13.1
9	25	12.5	45.4	**11.4**
10	22	45.5	**41.1**	48.9
11	22	3.8	47.2	3.9
12	24	3.7	31.8	4.8
13	28	30.0	31.3	34.0
14	28	6.1	25.8	**5.4**
15	22	**5.4**	27.4	6.2
16	28	2.4	14.9	2.6
17	23	25.3	23.8	31.0
18	26	8.6	38.3	8.6
19	22	22.7	23.6	24.7
20	20	8.6	29.7	**7.7**
Mean		20.45	35.71	21.57

dataset we may find some unexpected outcome. In this case, farthest-first generally outperforms d-Confidence in finding rare instances, contrary to what happens in text corpora. This is probably a sign that distance might be a better discriminator in low-dimensional input spaces than it is in high-dimensional input spaces.

Distance functions might lose their usefulness in high-dimensional spaces where the distance to the nearest and farthest neighbors come very similar—the curse-of-dimensionality [7]. This effect is most noticeable when using L_k-norm distances with a high value of k ($k \geq 3$). Euclidean distance, a L_2-norm metric, is not much affected [3]. To assess this effect on our datasets we have computed the *relative contrast*—measuring the relative distance of the nearest and farthest neighbors of a given query—for all instances in each dataset (4). In Table 12[2] we can observe that the discrimination between the nearest and farthest neighbors is not too sensitive to the data dimensionality. Despite the fact that the minimum relative contrast exhibits a negative correlation of 64 % to the data dimensionality, the maximum relative contrast is not correlated and there is no evidence

[2]Notation: *min.rc* and *min.rc* stand for the minimum and maximum relative contrast observed in each dataset; *global.rc* is a global contrast measure for each dataset computed by (4) but using the maximum and minimum distances between all the instances in the dataset.

Table 10 First-hit (ff-fh, c-fh and dc-fh) for the R52 dataset

Class	Freq	ff-fh	c-fh	dc-fh
1	239	1.0	24.0	1.0
2	5	78.5	115.6	**64.7**
3	3	230.3	**118.6**	178.7
4	2	**98.7**	167.4	107.8
5	6	239.0	173.7	**110.6**
6	11	7.5	80.0	10.0
7	4	15.9	123.6	19.1
8	3	130.0	173.3	**102.9**
9	7	240.2	128.8	136.0
10	2	153.2	118.0	**99.5**
11	40	14.6	12.4	20.0
12	2	209.9	158.5	166.4
13	435	2.5	25.2	4.0
14	2	219.0	152.2	150.4
15	3	192.8	214.1	**123.9**
16	7	113.7	**91.9**	107.8
17	9	**33.1**	92.7	46.3
18	5	24.9	96.7	**16.8**
19	2	**93.1**	140.0	104.7
20	3	411.8	206.7	**184.9**
21	2	273.2	143.6	154.5
22	2	588.6	**188.9**	202.8
23	30	76.0	**28.9**	63.4
24	4	341.9	171.7	171.1
25	4	253.9	**196.2**	224.0
26	2	459.6	313.1	**256.4**
27	5	282.8	**130.0**	150.7
28	2	294.7	216.3	**144.5**
29	2	422.5	**175.5**	198.7
30	3	**68.5**	213.3	85.2
31	2	**111.7**	206.0	126.7
32	2	248.3	233.7	**167.0**
33	30	53.0	**39.7**	49.7
34	15	67.6	**44.6**	99.0
35	4	**187.8**	271.6	219.6
36	2	**58.2**	153.2	84.1
37	3	45.7	137.6	44.8
38	3	66.6	159.3	**52.1**
39	2	101.2	226.0	106.9
40	2	90.4	144.3	**75.5**
41	5	67.6	68.7	62.9
42	3	206.6	159.1	**144.8**
43	4	43.4	153.4	**36.7**
44	14	72.7	103.8	76.6
45	3	**86.5**	179.7	123.9
46	12	3.2	68.6	6.6
47	2	45.9	148.5	51.1
48	3	101.9	160.8	**76.1**
49	35	39.4	36.4	72.9
50	3	219.0	175.6	**108.7**
51	3	482.2	**146.1**	183.5
52	2	302.7	258.8	**196.5**
Mean		159.10	143.58	107.16

that high-dimensional data are affecting the distance metric in use. The lack of correlation between the global contrast measure and data dimensionality supports this conclusion.

$$\frac{D_{\text{Max}} - D_{\text{min}}}{D_{\text{min}}} \tag{4}$$

At the R52 dataset the difference of mean error is significant in favor of confidence. D-Confidence reduces the labeling effort that is required to identify instances in R52, exhibiting better representativeness capabilities in this corpus. However, the error rate gets worse. Apparently, d-Confidence gets to know more classes from the target concept earlier although less sharply. In the R52 dataset we are exchanging accuracy for representativeness.

A similar analysis on the LDC for text corpora (Table 11) is not as clear on d-Confidence improvement. D-Confidence outperforms confidence on the R52 corpus, with a lower LDC by 22 % but confidence outperforms d-Confidence on NG, with a lower LDC by 34 %. Nevertheless it is relevant that d-Confidence, once again, performs better on imbalanced data.

Table 11 LDC for text corpora

Dataset	Classifier	ff.ldc	c.ldc	dc.ldc	Best
NG	SVM	87.9	63.7	85.4	c
R52	SVM	588.6	313.1	256.4	dc

Table 12 Relative contrast using Euclidean distance

Dataset	Dim.	Instances	min.rc	max.rc	global.rc
Iris	4	150	4.477	63.985	69.852
Vowels	10	330	2.424	51.371	65.253
Poker	10	500	2.117	7.426	9.630
Cleveland	13	298	1.085	59.944	68.921
Satlog	36	500	1.682	23.760	26.258
R52	6019	1000	0.315	52.292	67.812
NG	10333	500	0.428	23.034	32.197

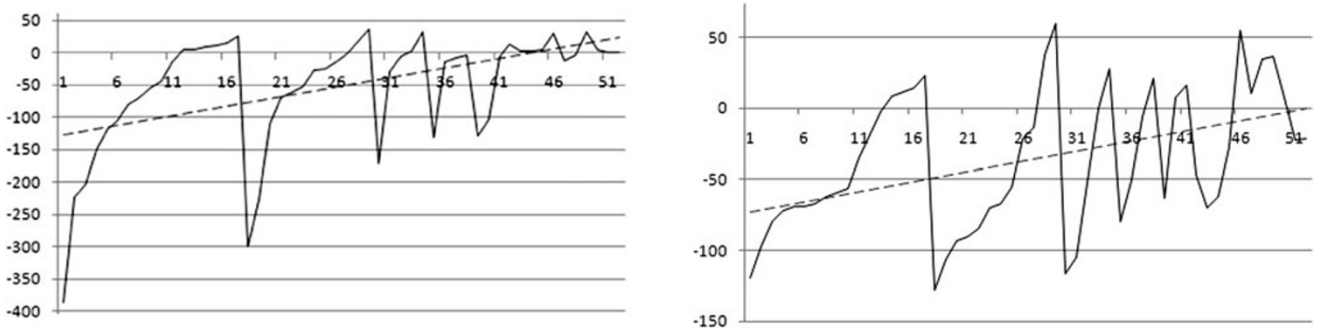
Figures 8 provide additional evidence on the ability of d-Confidence to find rare instances. These charts, where classes are sorted by increasing frequency, show that d-Confidence ensures a significant reduction in the mean number of queries that are required to first hit classes in R52. This reduction is more important in minority classes, i.e., in the first classes appearing in the horizontal axis. These charts represent the difference in d-Confidence first-hit compared to their baseline criteria. Negative differences mean that d-Confidence performed better, i.e., found representative instances of the class with fewer queries than its baseline criteria.

The dashed trend lines represented in both charts (Fig. 8), with a positive slope clearly show that the gain in d-Confidence first-hit, when compared to its both baseline criteria, decreases when the class frequency increases.

Another perspective of these results may clarify our point of view. In Fig. 9a we give, for each different value of class frequency in the working set, the number of classes that were first found by each criteria—lowest first-hit among all criteria. Figure 9b represents the accumulated number of first found classes. As detailed below, both these charts show evidence on the improved ability of d-Confidence to find exemplary instances of under-represented classes.

When comparing d-Confidence against farthest-first we can observe that from the 17 classes in R52 that have a frequency of 2, d-Confidence finds 11 before farthest-first. From the 12 classes with a frequency of 3, d-Confidence finds 10 before farthest-first. From the 13 classes with frequency between 4 and 9, d-Confidence finds 10 with fewer queries than farthest-first. From the remaining 10 classes, with a frequency between 11 and 435, d-Confidence finds only two before farthest-first.

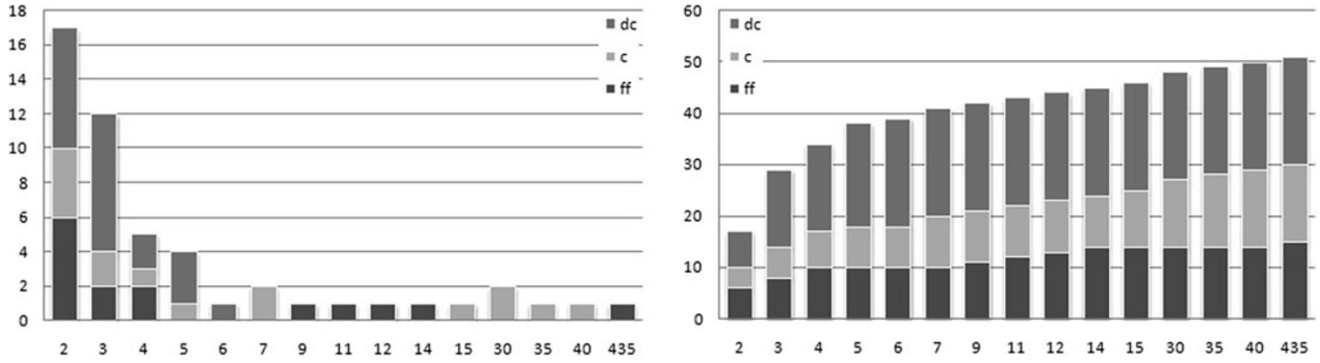
A similar comparison against confidence shows similar results. From the 17 classes in R52 that have a frequency



(a) D-Confidence vs farthest-first

(b) D-Confidence vs confidence

Fig. 8 Average gain of d-Confidence over its baseline criteria to first hit classes on R52. Classes are sorted by increasing frequency



(a) First found classes

(b) Accumulated first found

Fig. 9 Number of classes of a given frequency first found by each criteria on R52

of 2, d-Confidence finds 13 before confidence. From the 12 classes with a frequency of 3, d-Confidence finds 10 before confidence. From the 13 classes with frequency between 4 and 9, d-Confidence finds 10 with fewer queries than confidence. From the remaining 10 classes, with a frequency between 11 and 435, d-Confidence finds five before confidence.

4.7 Prevailing outcomes

The experimental results from both phases provide evidence on the performance of d-Confidence towards our objectives (Sect. 4).

Base classifier The performance of d-Confidence seems to be slightly affected by the base classifier, mainly w.r.t. error. When referring to known classes, d-Confidence generally improves over its base classifiers. D-Confidence is suited for SVM classifiers where it generally improves over its baseline criteria. When using other base classifiers the performance is affected but improvements are still observable.

Confidence vs. d-Confidence If we focus on SVM, we can observe that d-Confidence performs better than confidence, both at labeling effort and accuracy, over tabular datasets as well as over text corpora. D-Confidence dominates confidence w.r.t. known classes throughout all the learning process. D-Confidence also outperforms confidence first-hit performance, in general. This dominance is also evident w.r.t. error except on highly imbalanced datasets where confidence takes the lead.

Farthest-first vs. d-Confidence D-Confidence clearly dominates farthest-first w.r.t. error when using SVM classifiers. The relative performance of these two criteria when it comes to known classes depends on the class distribution at the working set. On balanced datasets, d-Confidence clearly outperforms farthest-first. On imbalanced datasets d-Confidence still outperforms farthest-first on average; however farthest-first generally beats d-Confidence in finding majority classes.

Data dimensionality The dimensionality of the input feature space does not compromise d-Confidence that exhibits performance improvements over its baseline criteria at tabular low-dimensionality data as well as at high-dimensional text corpora. However, some experimental results show that, unexpectedly, farthest-first outperforms d-Confidence when finding rare instances in imbalanced low-dimensional data (Poker dataset) while the same is not observed in high-dimensional data (R52 corpus). This has probably to do with the better discriminative abilities of distance at low-dimensional input spaces when compared to high-dimensional input spaces. This might require a parameter to tune the relative weight of confidence and distance in d-Confidence.

Balanced vs. imbalanced class distributions In general, d-Confidence outperforms its baseline criteria in finding exemplary instances from all the target classes. The gain is particularly relevant when finding under-represented classes in presence of highly imbalanced data. This gain however is achieved at the cost of accuracy. When in presence of imbalanced data, the exploratory bias of d-Confidence promotes exchanging accuracy for representativeness.

5 Conclusions and future work

The evaluation procedure that we have performed provided statistical evidence on the performance of d-Confidence when compared to its baseline criteria—confidence and farthest-first. D-Confidence reduces the labeling effort and

identifies exemplary cases for all classes faster than confidence and farthest-first alone. This gain is higher for minority classes, which are the ones where the benefits of d-Confidence become more relevant.

The base classifier used in the learning process has some influence on accuracy but apparently not on the labeling effort. D-Confidence consistently presents lower label disclosure complexity irrespectively of the base classifier. When it comes to error, the models generated by SVM classifiers seem to take better advantage of d-Confidence than neural networks or decision trees.

D-Confidence performs better in imbalanced datasets where it provides significant gains that greatly reduce the labeling effort. However, d-Confidence consistently outperforms confidence and farthest-first in terms of label complexity.

In general, d-Confidence improves the performance of its baseline criteria both from the exploration point of view—finding unknown classes faster—and from the exploitation point of view—improving, although marginally, the accuracy—when applied to tabular, low-dimensional data.

When applied to text corpora, farthest-first was outperformed by d-Confidence on the imbalanced corpus and presented similar performance on the balanced corpus, in terms of finding unknown classes, but with lower accuracy.

In general, d-Confidence achieved better performance on the imbalanced corpus than on the balanced one. The main drawback of d-Confidence when applied on the imbalanced text corpus is that the reduction in the labeling effort that is achieved in identifying unknown classes is obtained at the cost of increasing error. This increase in error is probably due to the fact that we are diverting the classifier from focusing on the decision function of the majority classes to focus on finding new, minority, classes. As a consequence the classification model generated by d-Confidence is able of identifying more distinct classes faster but gets less sharp in each one of them. This is particularly harmful for accuracy since a fuzzier decision boundary for majority classes might cause many erroneous guesses with a negative impact on error.

We are now exploring semi-supervised learning to leverage the intrinsic value of unlabeled instances so we can benefit from the reduction in labeling effort provided by d-Confidence and improve accuracy.

Comparing the instances that are being selected by each active learning strategy—for instance, by computing the percentage and class distribution of common selected instances as the learning process evolves—might help understanding operating patterns from each strategy. Although this line of work is in progress, the preliminary results reveal the distinction between confidence and farthest-first strategies.

Calculating distances between documents may be demanding and cause other limitations to d-Confidence. This effort can be reduced by first pre-selecting a subset of documents using a less demanding process and only then choosing the document to label. This is another line of future work.

Another fundamental aspect of active learning that we are focused on is the definition of a stopping criteria so we can decide when to stop querying.

References

1. Uc irvine machine learning repository (2009). http://archive.ics.uci.edu/ml/
2. Adami G, Avesani P, Sona D (2005) Clustering documents into a web directory for bootstrapping a supervised classification. Data Knowl Eng 54:301–325
3. Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional spaces. In: Proceedings of the 8th international conference on database theory, ICDT'01. Springer, London, pp 420–434. http://dl.acm.org/citation.cfm?id=645504.656414
4. Angluin D (1988) Queries and concept learning. Mach Learn 2:319–342. doi:10.1007/BF00116828
5. Balcan MF, Beygelzimer A, Langford J (2006) Agnostic active learning. In: ICML, pp 65–72.
6. Baum E (1991) Neural net algorithms that learn in polynomial time from examples and queries. IEEE Trans Neural Netw 2:5–19
7. Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton
8. Bonwell CC, Eison JA (1991) Active learning: creating excitement in the classroom. Jossey-Bass, San Francisco
9. Brinker K (2003) Incorporating diversity in active learning with support vector machines. In: Proceedings of the twentieth international conference on machine learning
10. Chakrabarti S (2002) Mining the Web: discovering knowledge from hypertext data. Morgan Kauffman, San Mateo. http://www.cse.iitb.ac.in/~soumen/mining-the-web/
11. Chapelle O, Schoelkopf B, Zien A (eds) (2006) Semi-supervised learning. MIT Press, Cambridge
12. Cohn D, Atlas L, Ladner R (1990) Training connectionist networks with queries and selective sampling. In: Advances in neural information processing systems
13. Cohn D, Atlas L, Ladner R (1994) Improving generalization with active learning. Mach Learn 15:201–221. doi:10.1023/A:1022673506211. http://portal.acm.org/citation.cfm?id=189256.189489
14. Cohn D, Ghahramani Z, Jordan M (1996) Active learning with statistical models. J Artif Intell Res 4:129–145
15. Dasgupta S (2005) Coarse sample complexity bonds for active learning. In: Advances in neural information processing systems, p 18
16. Dasgupta S, Hsu D (2008) Hierarchical sampling for active learning. In: Proceedings of the 25th international conference on machine learning
17. Escudeiro N, Jorge A (2006) Semantics, web and mining. In: Semi-automatic creation and maintenance of web resources with web Topic. LNCS, vol 4289. Springer, Heidelberg, pp 82–102
18. Escudeiro N, Jorge A (2008) Learning partially specified concepts with d-confidence. In: Brazilian symposium on artificial intelligence, web and text intelligence workshop
19. Escudeiro N, Jorge A (2009) Efficient coverage of case space with active learning. In: Lopes LS, Lau N (eds) Progress in artificial intelligence, proceedings of the 14th Portuguese conference

on artificial intelligence (EPIA 2009), vol 5816. Springer, Berlin, pp 411–422
20. Escudeiro N, Jorge AM (2010) D-Confidence: an active learning strategy which efficiently identifies small classes. In: Proceedings of the NAACL HLT 2010 workshop on active learning for natural language processing, association for computational linguistics, Los Angeles, CA, pp 18–26. http://10.255.0.115/pub/2010/EJ10
21. Escudeiro N, Jorge AM (2010) Reducing label complexity in the presence of imbalanced class distributions. In: Proceedings of the III international workshop on web and text intelligence (WTI—2010), São Bernardo do Campo, São Paulo, Brazil. http://10.255.0.115/pub/2010/EJ10a
22. Hanneke S (2007) A bound on the label complexity of agnostic active learning. In: Proceedings of the 24th international conference on machine learning
23. Hochbaum D, Shmoys D (1985) A best possible heuristic for the k-center problem. Math Oper Res 10(2):180–184
24. Hoi S, Jin R, Lyu M (2006) Large-scale text categorization by batch mode active learning. In: Proceedings of the world wide web conference
25. Hoi SCH, Jin R, Zhu J, Lyu MR (2009) Semisupervised svm batch mode active learning with applications to image retrieval. ACM Trans Inf Syst 27(3):1–29. doi:10.1145/1508850.1508854
26. Hu W, Hu W, Xie N, Maybank S (2009) Unsupervised active learning based on hierarchical graph-theoretic clustering. Trans Syst Man Cybern, Part B 39(5):1147–1161. doi:10.1109/TSMCB.2009.2013197
27. Huang A, Milne D, Frank E, Witten IH (2008) Clustering documents with active learning using Wikipedia. In: ICDM'08: proceedings of the 2008 eighth IEEE international conference on data mining. IEEE Comput. Soc., Washington, pp 839–844. doi:10.1109/ICDM.2008.80
28. Kääriäinen M (2006) Active learning in the non-realizable case. In: Algorithmic learning theory. Springer, Berlin/Heidelberg, pp 63–77
29. Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: SIGIR'94: proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval. Springer, New York, pp 3–12
30. Li M, Sethi I (2006) Confidence-based active learning. IEEE Trans Pattern Anal Mach Intell 28:1251–1261
31. Liu H, Motoda H (2001) Instance selection and construction for data mining. Kluwer Academic, Dordrecht
32. Mitchell TM (1997) Machine learning. McGraw-Hill, New York
33. Muslea I, Minton S, Knoblock CA (2006) Active learning with multiple views. J Artif Intell Res 27:203–233
34. Nguyen HT, Smeulders A (2004) Active learning using pre-clustering. In: Proceedings of the 21st international conference on machine learning. ACM, New York, pp 623–630
35. Ribeiro P, Escudeiro N (2008) On-line news "à la carte". In: Proceedings of the European conference on the use of modern information and communication technologies
36. Roy N, McCallum A (2001) Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of the eighteenth international conference on machine learning, ICML'01. Morgan Kaufmann, San Francisco, pp 441–448. http://portal.acm.org/citation.cfm?id=645530.655646
37. Schohn G, Cohn D (2000) Less is more: active learning with support vector machines. In: Proceedings of the international conference on machine learning
38. Seung H, Opper M, Sompolinsky H (1992) Query by committee. In: Proceedings of the 5th annual workshop on computational learning theory