Available online at www.sciencedirect.com

SciVerse ScienceDirect

Int. J. Human-Computer Studies 70 (2012) 321–331

International Journal of Human-Computer Studies

www.elsevier.com/locate/ijhcs

Evaluation methods and strategies for the interactive use of classifiers

Silvia Acid, Luis M. de Campos*, Moisés Fernández

Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática y de Telecomunicación, CITIC-UGR Universidad de Granada, 18071 Granada, Spain

Received 4 October 2010; received in revised form 26 October 2011; accepted 9 January 2012
Communicated by M. Zanker
Available online 21 January 2012

Abstract

We consider the scenario in which an automatic classifier (previously built) is available. It is used to classify new instances but, in some cases, the classifier may request the intervention of a human (the oracle), who gives it the correct class. In this scenario, first it is necessary to study how the performance of the system should be evaluated, as it cannot be based solely on the predictive accuracy obtained by the classifier but it should also take into account the cost of the human intervention; second, studying the concrete circumstances under which the classifier decides to query the oracle is also important. In this paper we study these two questions and include also an experimental evaluation of the different proposed alternatives.

Keywords: Interactivity; Classification; Uncertainty sampling; Evaluation models

1. Introduction

Most of the work in automatic classification is focused in the construction of the classifier from data (using either a supervised, unsupervised, semi-supervised, active or even online strategy) and no attention is paid to how to use it later, in a realistic situation.

We consider the scenario where the classifier has already been built and now we are going to use it. In such a case, we can let the classifier to do its job autonomously and accept its predictions (which sometimes may be wrong) or we could also interact with the classifier in order to improve the performance (Stumpf et al., 2009). The simple kind of interaction considered is that the classifier, when faced with a new instance to classify, can decide either to classify it (correctly or wrongly) or to query a human supervisor (the oracle) the true class label for this instance. We may call this task interactive use of a classifier or, for short, interactive classification. This should not be confused with the concept of interactive machine learning (Ware et al., 2001), where users actually generate classifiers themselves with the help of two-dimensional visual interfaces.

Notice that this situation shares some similarities with the active learning (Cohn et al., 1994; Lewis and Gale, 1994) and the online learning (Littlestone, 1988; Littlestone and Warmuth, 1994; Helmbold and Panizza, 1997) approaches to build classifiers but the goal is different. In active learning the classifier asks the oracle the true class labels of some selected instances, in order to iteratively build a good classifier using less training data. However, in our case we do not want to build the classifier (it has already been built), we cannot control what instances to use (we only receive new instances that need to be classified) and, in principle, we do not try to improve it.[1] On the other hand, the key defining characteristic of online learning is that soon after the prediction of the classifier is made, the true label of the new instance will be known. For example, in any problem that consists of predicting the future, an online learning algorithm just needs to wait for the label to become available.

*Corresponding author. Tel.: +34 958243199; fax: +34 958243317.
E-mail addresses: acid@decsai.ugr.es (S. Acid), lci@decsai.ugr.es (L.M. de Campos), moises@decsai.ugr.es (M. Fernández).

[1] Although we could also try to use the new information provided by the oracle to refine the classifier, here we do not consider this problem, as it depends on the specific classifier being used and in this paper we are focused on general evaluation methods.

1071-5819/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijhcs.2012.01.002

This information can then be used to refine the prediction capabilities of the classifier. The scenario for our interactive classification task is different, a scenario where the true label of the new instance will not be known for sure (and therefore we take the risk of permanently accepting a wrong classification) unless we explicitly query the oracle. An example of this situation can be the classification of text documents (Sebastiani, 2002) in a set of categories (e.g. a taxonomy of web directories, Dumais and Chen, 2000, a controlled vocabulary, Golub, 2006 or a thesaurus, de Campos and Romero, 2009).

In this scenario, first it is necessary to study how the performance of the system should be evaluated, as it cannot be based solely on the predictive accuracy obtained by the classifier but should also take into account the cost of the human intervention. For example a system that always queries the oracle will obtain an accuracy of 100%, but it is a useless system (the rate of human intervention is also 100%). On the contrary, a system that never queries (and is not perfect) could be outperformed by an interactive classifier that occasionally asks labels to an oracle. We will define evaluation methods for the interactive classification task.

Another important issue is the way in which the system decides whether or not to request a label to the oracle. A reasonable approach is to query when the classifier has little confidence in the correctness of its own prediction. For example, in a probabilistic model, the decision could be based on the shape of the posterior distribution of the class variable. Query strategies from the stream-based active learning literature (Settles, 2009) could be considered or adapted.

The remainder of this paper is organized in the following way: we first study in Section 2 evaluation criteria for the interactive classification problem. In Section 3 we consider strategies for deciding when the interactive classifier should query the oracle. Section 4 contains an experimental evaluation of the different proposed alternatives. Finally, Section 5 includes the conclusions and some proposals for future research.

2. Evaluation criteria for the interactive classification problem

Consider the problem of classifying each one of a set of N instances into one of the possible labels or categories in $C=\{c_1,\ldots,c_m\}$, and assume that we have already built a classifier (using a set of training instances and a learning algorithm). Assume also that the classifier, in addition to predict the category of an example, can alternatively ask its true label (using some strategy as those considered in Section 3). Let n_c and n_w be the number of times that the classifier gets the correct and the wrong category, respectively, and let n_i be the number of interactions, i.e. the number of times that the classifier decided to ask the oracle instead of trusting its own prediction. Obviously $n_c+n_w+n_i=N$.

The question that we consider here is how can we evaluate the quality of this interactive classifier. For example, if $N=100$ and using a decision strategy and a given classifier we obtain $n_c^1=60$, $n_w^1=10$ and $n_i^1=30$, whereas using another strategy and/or another classifier we get $n_c^2=69$, $n_w^2=16$ and $n_i^2=15$, which of these two situations is preferable?

We want to define an evaluation measure that takes into account not only the number of correctly and wrongly classified examples (as the predictive accuracy, $n_c/(n_c+n_w)$, in a non-interactive scenario) but also the number n_i of interactions with the oracle. To this end, we are going first to specify several properties that such an evaluation measure reasonably should possess.

Let $f(n_c,n_w,n_i)$ be an evaluation measure for interactive classification (EMIC), whose value we want to maximize. The following properties should be satisfied by any reasonable EMIC:

1. To increase the number of examples correctly classified by the classifier while decreasing the number of examples wrongly classified (and keeping unmodified the number of interactions) is beneficial: $\forall k$ such that $1\le k\le n_w$,
$f(n_c,n_w,n_i)<f(n_c+k,n_w-k,n_i)$.

2. To increase the number of examples correctly classified, at the expense of decreasing the number of interactions is also beneficial: $\forall k$ such that $1\le k\le n_i$,
$f(n_c,n_w,n_i)<f(n_c+k,n_w,n_i-k)$

3. To increase the number of interactions is positive if we reduce the number of examples wrongly classified: $\forall k$ such that $1\le k\le n_w$,
$f(n_c,n_w,n_i)<f(n_c,n_w-k,n_i+k)$

4. To increase the number of correctly classified examples increases the performance: $\forall k>0$,
$f(n_c,n_w,n_i)\le f(n_c+k,n_w,n_i)$

5. To increase the number of wrongly classified examples decreases the performance: $\forall k>0$,
$f(n_c,n_w,n_i)>f(n_c,n_w+k,n_i)$

We shall say that an EMIC satisfying these five properties is a *coherent* EMIC. Observe that, as a consequence of the first two properties we have that the best possible result is to always classify correctly without requiring any interactions, i.e.

$$\max_{n_c,n_w,n_i|n_c+n_w+n_i=N} f(n_c,n_w,n_i)=f(N,0,0)$$

It should also be noticed that from the third property it follows immediately that $f(n_c,n_w+n_i,0)<f(n_c,n_w,n_i)<f(n_c,0,n_i+n_w)$. This means that the optimal strategy to

query labels, which consists in querying only when the classifier is going to make a mistake, is always preferable to the non-interactive classifier. The value $f(n_c, 0, n_i+n_w)$ represents the upper limit of performance for a classifier whose (non-interactive) accuracy is n_c/N.

We can think in different ways of defining an EMIC. For example, it may be natural to assume that the quality of the interactive classifier depends on both the proportion of correctly assigned examples, $(n_c+n_i)/N$ (remember that the oracle always returns the true label) and its degree of autonomy (capacity of doing the job without querying the oracle), $(N-n_i)/N$. Then we can combine these two factors by using some kind of average. Thus we would obtain:

- Arithmetic mean: $(n_c+N)/2N$
- Geometric mean: $(1/N)\sqrt{(n_c+n_i)(N-n_i)}$
- Harmonic mean: $2(n_c+n_i)(N-n_i)/N(n_c+N)$

However, none of these three measures is a coherent EMIC, since they all fail to satisfy the property 3 above. Consider for example the case where $n_c=50$, $n_w=40$, $n_i=10$ and $k=40$: for all the three cases we get $f(50,40,10) \geq f(50,0,50)$.

Another option is to base our measure in the concepts of precision and recall[2] (which are so frequently used in the information retrieval and text classification fields). Let us define the precision p as the ratio between the number of examples correctly classified and the number of examples *effectively* classified by the classifier alone (without interactions), i.e. $p=n_c/(n_c+n_w)=n_c/(N-n_i)$. The recall r is defined as the ratio between the number of examples correctly classified and the total number of examples, $r=n_c/(n_c+n_w+n_i)=n_c/N$. The standard way of combining these two measures is through the use of the so called F_β measure (van Rijsbergen, 1979), which is the weighted harmonic mean of precision and recall: $F_\beta=(1+\beta^2)pr/(\beta^2 p+r)$. The range of F_β is [0,1] and the parameter β varies in $(0,+\infty)$; when $\beta=1$ we are giving the same weight to precision and recall; using $\beta>1$ we are giving more weight to recall, whereas with $\beta<1$ we give more importance to precision. In our case, the F_β measure is the following:

$$F_\beta=\frac{(1+\beta^2)pr}{\beta^2 p+r}=\frac{(1+\beta^2)n_c}{(1+\beta^2)N-n_i} \tag{1}$$

Observe that in the non-interactive case, i.e. when $n_i=0$, the value of the measure is $F_\beta=n_c/N$, so we reproduce the accuracy.

Proposition 1. *The F_β measure defined in Eq. (1) is a coherent EMIC.*

The proof is very simple and we omit the details. All the five properties defining a coherent EMIC can be easily proven by induction on the number k, starting with $k=1$ and taking into account that $n_c+n_i \leq N$.

A second proposal for defining a coherent EMIC is by thinking in terms of cost. The three possible results that, given an instance to classify, the interactive classifier can return are classify correctly, classify wrongly and query. Let d_w and d_i be the costs incurred by classifying wrongly and asking, respectively; the cost of classifying correctly will obviously be $d_c=0$. Then, we can evaluate the interactive classifier by estimating the average cost (AC) in the following way:

$$AC=d_c\frac{n_c}{N}+d_w\frac{n_w}{N}+d_i\frac{n_i}{N}=d_w\frac{N-n_c-n_i}{N}+d_i\frac{n_i}{N}$$
$$=d_w\frac{N-n_c}{N}+(d_i-d_w)\frac{n_i}{N}$$

It is quite natural to require that the cost of a wrong classification should be greater than the cost of querying, i.e. $d_w>d_i$. Otherwise an interactive classifier would have not any sense because the cheapest classifier would always be the non-interactive one.

As only the proportion between d_i and d_w is important in terms of comparing classifiers and/or query strategies, and in order to manage only one parameter, we define $\rho=d_i/d_w$ and then redefine the average cost by also dividing by d_w, thus obtaining

$$AC_\rho=\frac{1}{d_w}\left(d_w\frac{N-n_c}{N}+(d_i-d_w)\frac{n_i}{N}\right)=\frac{N-n_c-(1-\rho)n_i}{N}$$

It should be noticed that an EMIC is defined in such a way that the greater the value the better the performance, and with the average cost the opposite situation occurs. So, we are going to speak in terms of profit instead of cost to define the EMIC, simply by subtracting the average cost from the maximum cost (which is equal to 1). The expected profit EP_ρ is then defined as

$$EP_\rho=1-\frac{N-n_c-(1-\rho)n_i}{N}=\frac{n_c+(1-\rho)n_i}{N} \tag{2}$$

The range of the EP_ρ measure is also the interval [0,1] and its value in the non-interactive case ($n_i=0$) is again equal to the accuracy, $EP_\rho=n_c/N$.

Proposition 2. *The EP_ρ measure defined in Eq. (2) is a coherent EMIC if $0<\rho<1$.*

The proof of this result can be obtained by simple algebraic manipulations, so that we omit it. The parameter ρ in Eq. (2) must be understood as the ratio between the cost of querying and the cost of a wrong classification. It can also be interpreted as the difference between the utility of classifying correctly (which is equal to 1) and the utility of asking the oracle (assuming that the utility of a wrong classification is zero).

[2]More precisely, we shall consider the concepts of *micro precision* and *micro recall* (Sebastiani, 2002).

3. Decision strategies to ask the oracle

In this section we study several strategies that the interactive classifier could use in order to decide when it should not classify a new instance and instead it should ask the oracle. In the absence of any kind of external information, the only reasonable approach is to base the decision on the degree of confidence of the classifier in its own prediction. We shall assume that, given an instance $\mathbf{x}$, the classifier obtains a set of numerical values representing the degree of confidence of the classifier in that each one of the categories c_i is the true category for this instance. These values can be interpreted as the posterior probability distribution of the class variable given the instance,[3] $p(C|\mathbf{x})$. The classifier would then predict the most probable class, $c^*_{\mathbf{x}} = \arg\max_{c_i} p(c_i|\mathbf{x})$.

In the active learning literature a very similar problem has already been considered, namely that of selecting the most informative instances that must be queried (those that would probably improve the classification model the most). In the pool-based active learning setting the queries are selected from a large pool of unlabeled instances, whereas in the stream-based active learning setting only one instance is available each time and then the learner has to decide whether or not to ask its label. In any case this involves evaluating in some way the informativeness of unlabeled instances. Although in our case the goal is different (confidence versus informativeness), some of the query strategies for active learning can also be useful here, concretely the so-called uncertainty sampling (Lewis and Gale, 1994), which selects unlabeled examples for querying, based on the level of uncertainty about their correct class: the active learner queries the instances about which it has the least certainty.

There are several ways of measuring the degree of confidence of the classifier that we shall describe below. Let $\phi(p)$ be the confidence degree in the prediction obtained from the posterior probability $p(C|\mathbf{x})$. By fixing a minimum threshold, α, on the degree of confidence, we can establish a decision rule according to which the system will decide whether or not to query the oracle:

$$\begin{aligned}&\text{if } \phi(p) \le \alpha \text{ then ask the oracle}\\ &\text{else classify the instance}\end{aligned} \tag{3}$$

The case where the classifier has maximum confidence in its prediction is, clearly, when $p(c^*_{\mathbf{x}}|\mathbf{x}) = 1$ and $p(c_i|\mathbf{x}) = 0$ for all $c_i \neq c^*_{\mathbf{x}}$. The different measures of confidence $\phi(p)$ considered are the following:

- Maximum probability:

$$\phi_m(p) = \max_{c_i} p(c_i|\mathbf{x}) = p(c^*_{\mathbf{x}}|\mathbf{x}) \tag{4}$$

  $\phi_m(p)$ varies in the range $[1/m, 1]$. $\phi_m(p) = 1/m$ represents absolute lack of confidence (because in this case $p(c_i|\mathbf{x}) = 1/m$ for all c_i), whereas $\phi_m(p) = 1$ means total confidence.

- Difference between the two most probable class labels:

$$\phi_d(p) = p(c^*_{\mathbf{x}}|\mathbf{x}) - p(c^{**}_{\mathbf{x}}|\mathbf{x}) \tag{5}$$

  where $c^{**}_{\mathbf{x}} = \arg\max_{c_i \neq c^*_{\mathbf{x}}} p(c_i|\mathbf{x})$. $\phi_d(p)$ varies in the range [0,1], with $\phi_d(p) = 1$ meaning again total confidence and $\phi_d(p) = 0$ representing complete distrust ($p(c^*_{\mathbf{x}}|\mathbf{x}) = p(c^{**}_{\mathbf{x}}|\mathbf{x})$), because there are at least two competing class labels having the same maximum probability.

- Negative entropy[4]

$$\phi_e(p) = \sum_{i=1}^{m} p(c_i|\mathbf{x}) \log(p(c_i|\mathbf{x})) \tag{6}$$

  $\phi_e(p)$ varies in the range $[-\log(m), 0]$, where $\phi_e(p) = 0$ means in this case total confidence and $\phi_e(p) = -\log(m)$ means total distrust ($p(c_i|\mathbf{x}) = 1/m$ for all c_i).

- Sample standard deviation (of the posterior probabilities):

$$\phi_s(p) = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} \left(p(c_i|\mathbf{x}) - \frac{1}{m} \right)^2} \tag{7}$$

  $\phi_s(p)$ varies in the range $[0, 1/\sqrt{m}]$, and again $\phi_s(p) = 0$ represents lack of confidence ($p(c_i|\mathbf{x}) = 1/m$ for all c_i) and $\phi_s(p) = 1/\sqrt{m}$ total confidence.

- Euclidean distance:

$$\phi_{ed}(p) = \sqrt{\left(\frac{p(c^*_{\mathbf{x}}|\mathbf{x})}{p(c^*_{\mathbf{x}}|\mathbf{x}) + p(c^{**}_{\mathbf{x}}|\mathbf{x})} - 0.5 \right)^2 + \left(\frac{p(c^{**}_{\mathbf{x}}|\mathbf{x})}{p(c^*_{\mathbf{x}}|\mathbf{x}) + p(c^{**}_{\mathbf{x}}|\mathbf{x})} - 0.5 \right)^2}$$

  This is the Euclidean distance between the vector composed of the two normalized maximum probabilities and the vector (0.5,0.5). Simple algebraic manipulation shows that $\phi_{ed}(p)$ can also be expressed as follows:

$$\phi_{ed}(p) = \frac{1}{\sqrt{2}} \frac{p(c^*_{\mathbf{x}}|\mathbf{x}) - p(c^{**}_{\mathbf{x}}|\mathbf{x})}{p(c^*_{\mathbf{x}}|\mathbf{x}) + p(c^{**}_{\mathbf{x}}|\mathbf{x})} \tag{8}$$

  $\phi_{ed}(p)$ varies in the range $[0, 1/\sqrt{2}]$ and, as in the previous cases, $\phi_{ed}(p) = 0$ represents maximum distrust ($p(c^*_{\mathbf{x}}|\mathbf{x}) = p(c^{**}_{\mathbf{x}}|\mathbf{x})$) and $\phi_{ed}(p) = 1/\sqrt{2}$ maximum confidence.

The criteria of maximum probability, difference between the two most probable class labels and entropy have already been used in the active learning literature (Settles, 2009; Settles and Craven, 2008), whereas sample standard deviation and Euclidean distance are new proposals. ϕ_s and ϕ_{ed} represent other ways of measuring the confidence in the prediction of the classifier, by evaluating the degree of 'flatness' or 'sharpness' of the posterior probability distribution. ϕ_s measures the sample standard

[3] The values will be conveniently normalized if necessary, in case of using a non probabilistic model.

[4] We use the negative of the entropy because the lower the entropy the greater the degree of confidence.

deviation from the probability values to its average (which is always equal to $1/m$). ϕ_{ed} is similar to ϕ_d, it evaluates the difference between the probabilities of the two most probable class labels but using another distance measure (Euclidean instead of absolute value) and normalizing the probabilities. ϕ_{ed} is also similar to ϕ_s, but computing the standard deviation only of the two most probable (normalized) probabilities.

It should be noticed that in the case of a binary classification problem ($m=2$), all these confidence measures become equivalent, in the sense that given a threshold α_r for any confidence measure ϕ_r, we can find another threshold α_s for any other confidence measure ϕ_s such that the decision strategies $\phi_r(p) \leq \alpha_r$ and $\phi_s(p) \leq \alpha_s$ are identical. For the non binary case, in general the different confidence measures do not generate the same decision strategies.

4. Experimental evaluation

We have carried out an extensive experimentation to study our interactive classification framework, using several classifiers and several databases. The main objectives are (1) testing whether the interactive version of the classifiers can be useful and (2) selecting the most promising confidence measures used by the decision strategy.

4.1. Databases and algorithms

We have selected six classifiers from the Weka platform (Hall et al., 2009): `HillClimber` (HC), `IBk`, `J48`, `Logistic` (LOG), `MultilayerPerceptron` (MP) and `NaiveBayes` (NB). `HillClimber` is a Bayesian network classifier (Heckerman et al., 1995); `IBk` is a *K*-Nearest Neighbors algorithm (Aha and Kibler, 1991); `J48` is the Weka version of the C4.5 decision tree algorithm (Quinlan, 1993); `Logistic` is a logistic regression classifier (Hilbe, 2009); `MultilayerPerceptron` is a feedforward artificial neural network model (Haykin, 1998), and `NaiveBayes` is self-explained. Observe that three algorithms (`HillClimber`, `Logistic` and `NaiveBayes`) are truly probabilistic classifiers, whereas the other three are not, but Weka is able to transform their predictions into probabilities. In all the cases we used the default options of these algorithms, except for `IBk`, where the number of neighbors was set to 3, and `HillClimber`, where the local search used the operators of arc addition, arc deletion and arc reversal to built the network structure, starting from an empty network and using the BDeu score.

We have also selected 31 databases from the UCI repository of machine learning databases (Blake and Merz, 1998). Table 1 gives a brief description of the characteristics of each database, including the number of instances, attributes and states for the class variable. These data sets have been preprocessed in the following way: the continuous variables were discretized using the procedure proposed by Fayyad and Irani (1993), and the instances with undefined/missing values were eliminated. For this preprocessing stage, we have used the MLC++ System (Kohavi et al., 1994).

Table 1
Description of the data sets used in our experiments.

#	Data set	Instances	Attributes	Classes
1	adult	45222	14	2
2	australian	690	14	2
3	breast	682	10	2
4	car	1728	6	4
5	chess	3196	36	2
6	cleve	296	13	2
7	corral	128	6	2
8	crx	653	15	2
9	diabetes	768	8	2
10	DNA-nominal	3186	60	3
11	flare	1066	10	2
12	german	1000	20	2
13	glass2	163	9	2
14	glass	214	9	7
15	heart	270	13	2
16	hepatitis	80	19	2
17	iris	150	4	3
18	letter	20000	16	26
19	lymphography	148	18	4
20	mofn-3-7-10	1324	10	2
21	mushroom	8124	22	2
22	nursery	12960	8	5
23	pima	768	8	2
24	satimage	6435	36	6
25	segment	2310	19	7
26	shuttle-small	5800	9	7
27	soybean-large	562	35	19
28	splice	3190	60	3
29	vehicle	846	18	4
30	vote	435	16	2
31	waveform-21	5000	21	3

The performance measures considered are the two coherent EMICs defined in Section 2, namely the F measure F_β and the expected profit EP_ρ, for different values of the parameters β and ρ. To estimate these measures we used 5-fold cross-validation, so that each time four fifths of the instances are used for training the base classifier and a fifth is used for testing.

In Fig. 1 we illustrate the experimental process globally: each of the six classifiers takes as the input each one of the instances (using cross validation) of each one of the 31 databases, and returns the posterior probability and the predicted class for each instance. This probability is used to compute a measure of the confidence in the prediction, according to each of the five different measures ϕ_r. Depending on the confidence value and the threshold α, the system decides either to query the oracle the true class label or to keep the original prediction. Finally, the labeled samples (from the test fold) are evaluated according to the proposed EMICs.

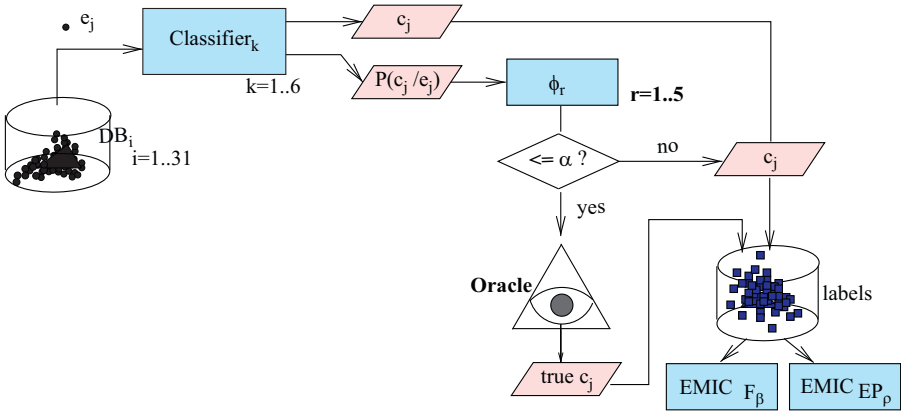

Fig. 1. Diagram of the experimental process.

4.2. Results

We have carried out different experiments, within the experimental setting previously described, in order to study different aspects of the interactive classification process. They are described in the following subsections.

4.2.1. Selection of the threshold α

A key parameter for the interactive classifier's decision strategy is the threshold α used in combination with the confidence measure: the greater α the more interactive the classifier. In the extreme case where α is set to its maximum value (which depends on the specific measure of confidence considered) then the classifier would always query; on the contrary, if α is set to its minimum value, then the classifier would be completely autonomous and it would never query.

To determine an upper limit of the performance of the interactive classifiers, first we are going to optimistically evaluate them using the best possible threshold (for each algorithm, each database and each confidence measure). To determine it, we rank the *test* instances (of each fold) in decreasing order of their confidence measure and then look for the threshold which optimizes the performance. We do it by tentatively making the threshold equal to the confidence measure of each test instance, computing the corresponding performance measure and selecting the best. A summary of the results of these experiments for the parameters $\beta = 0.5$ and $\rho = 0.5$, showing the averages across the 31 databases, is displayed in Table 2. As we can observe, we always get better results using the interactive classifiers, showing the potentiality of our approach, provided that we are able to find a good threshold. Although the differences with respect to the non-interactive classifier are not very large, they are in all the cases statistically significant (at level 0.05): for each base classifier, we first used the non-parametric Friedman test (more precisely the Iman–Davenport extension, Demsar, 2006) to reject the hypothesis that the results of the interactive and non-interactive classifiers are equal. Then, the Hommel post-hoc test (with the non-interactive classifier acting as the control method, Demsar, 2006) was used, again for each base classifier, and detected statistically significant differences with respect to the non-interactive classifier for all the interactive classifiers based on the five different confidence measures proposed.

Table 2
Macro-average values of the performance measures $F_{0.5}$ and $EP_{0.5}$ across the 31 databases, for each algorithm and each confidence measure, when computing the threshold α *optimistically*. All the differences with respect to the non-interactive classifier are statistically significant at level 0.05.

Algorithm	ϕ_e	ϕ_m	ϕ_d	ϕ_s	ϕ_{ed}	No interaction
	$F_{0.5}$					
HC	0.8978	0.8980	0.8977	0.8980	0.8974	0.8734
IBk	0.8891	0.8901	0.8898	0.8897	0.8894	0.8611
J48	0.8816	0.8820	0.8821	0.8821	0.8822	0.8654
LOG	0.8894	0.8893	0.8892	0.8894	0.8892	0.8685
MP	0.9029	0.9026	0.9026	0.9026	0.9026	0.8800
NB	0.8881	0.8884	0.8880	0.8881	0.8875	0.8551
	$EP_{0.5}$					
HC	0.8818	0.8818	0.8813	0.8819	0.8810	0.8734
IBk	0.8708	0.8718	0.8719	0.8712	0.8716	0.8611
J48	0.8710	0.8715	0.8715	0.8715	0.8712	0.8654
LOG	0.8753	0.8755	0.8753	0.8753	0.8752	0.8685
MP	0.8886	0.8884	0.8884	0.8884	0.8884	0.8800
NB	0.8675	0.8676	0.8672	0.8674	0.8669	0.8551

In order to determine a threshold in a realistic way, we proceed as follows: as before, we again look for the threshold which optimizes the performance, but now ranking (in decreasing order of their confidence measure) the *training* instances of each fold instead of the test instances. Then we apply this threshold to the test set and compute the performance. In this way we are not using the test instances to select the threshold. A summary of the results of these experiments appears in Table 3.

In this case the behavior of the two EMICs is somewhat different: using $F_{0.5}$ we still obtain always better results using the interactive classifiers, with differences statistically significant in all the cases; however, when using $EP_{0.5}$, we even get worse results in 9 from the 30 cases, and the differences are not statistically significant in any case, according to the Friedman test. It should be noticed that

Table 3
Macro-average values of the performance measures $F_{0.5}$ and $EP_{0.5}$ across the 31 databases, for each algorithm and each confidence measure, when computing the threshold α *realistically*. For $F_{0.5}$ all the differences with respect to the non-interactive classifier are statistically significant at level 0.05, whereas for $EP_{0.5}$ the differences are not significant.

Algorithm	ϕ_e	ϕ_m	ϕ_d	ϕ_s	ϕ_{ed}	No interaction
	$F_{0.5}$					
HC	0.8896	0.8892	0.8890	0.8886	0.8885	0.8734
IBk	0.8790	0.8804	0.8802	0.8801	0.8797	0.8611
J48	0.8737	0.8742	0.8736	0.8743	0.8735	0.8654
LOG	0.8796	0.8792	0.8793	0.8794	0.8793	0.8685
MP	0.8900	0.8904	0.8906	0.8905	0.8906	0.8800
NB	0.8761	0.8757	0.8758	0.8758	0.8749	0.8551
	$EP_{0.5}$					
HC	0.8757	0.8756	0.8749	0.8752	0.8746	0.8734
IBk	0.8626	0.8640	0.8637	0.8630	0.8642	0.8611
J48	0.8667	0.8669	0.8667	0.8668	0.8666	0.8654
LOG	0.8683	0.8683	0.8685	0.8683	0.8684	0.8685
MP	0.8768	0.8769	0.8770	0.8770	0.8769	0.8800
NB	0.8596	0.8595	0.8593	0.8595	0.8591	0.8551

the parameters $\beta = 0.5$ and $\rho = 0.5$ have a very different meaning, so that this different behavior is not necessarily odd.

4.2.2. Study of the EMIC parameters β and ρ

It is clear that the results obtained by the interactive classifiers will depend on the parameters β and ρ used by the two EMICs, which reflect (implicitly in the case of F_β and explicitly for EP_ρ) the relation between the costs of querying and misclassifying for a given problem and a given user. We have repeated the previous experiments (estimating the threshold α realistically) but using different values for ρ and β, more precisely ρ varying in $Z = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and β varying in $Z \cup \{1.0\}$.

Given the great volume of experiments, even showing a summary of the results (aggregated across the 31 databases), as we have done previously, is impractical. Therefore, in Tables 4 and 5 we only show results aggregated across the 31 databases, the 6 learning algorithms and the 5 confidence measures, representing the number of times that the interactive classifier obtains significantly better/better/worse/significantly worse results than the non-interactive classifier (with respect to F_β and EP_ρ, respectively). As before, we use the Friedman test to test the hypothesis that there are not differences between the interactive and non-interactive classifiers and, in case of rejection, the Hommel post-hoc test to detect significant differences with the non-interactive classifier.

It is clear from these results that the interactive classifiers should be used in combination with any base classifier only if the cost of a wrong classification is considered at least twice the cost of querying ($\rho \leq 0.5$, when using EP_ρ), or when we give more importance to precision than to recall ($\beta \leq 0.7$, if we are using F_β). As precision measures the performance of the classifier when it does not resort to the oracle, to give more importance to precision than to recall seem us quite natural, because obtaining low precision values means that a lot of examples would be misclassified. An example where these requirements are met could occur in a medical domain, where misclassifying an example may represent either not to give treatment to an ill patient or giving treatment (which in some cases may be quite aggressive) to a healthy patient. Another example could be an email classifier, where misfiling important messages could cause significant loses to the users (Stumpf et al., 2009).

Table 4
Number of times that the interactive classifier obtains significantly better/better/worse/significantly worse results (with respect to F_β) than the non-interactive classifier, from the 30 possible combinations of learning algorithms and confidence measures being considered, for different values of the parameter β.

β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Signific. better	30	30	30	30	30	30	15	0	0	0
Better	0	0	0	0	0	0	15	25	15	6
Worse	0	0	0	0	0	0	0	5	11	21
Signific. worse	0	0	0	0	0	0	0	0	4	3

Table 5
Number of times that the interactive classifier obtains significantly better/better/worse/significantly worse results (with respect to EP_ρ) than the non-interactive classifier, from the 30 possible combinations of learning algorithms and confidence measures being considered, for different values of the parameter ρ.

ρ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Signific. better	30	30	30	25	0	0	0	0	0
Better	0	0	0	5	21	8	3	0	0
Worse	0	0	0	0	9	18	17	18	20
Signific. worse	0	0	0	0	0	4	10	12	10

4.2.3. Comparing the confidence measures

Once we have shown that the use of interactive classifiers can be beneficial, we turn our attention to the selection of the best confidence measures. As all the confidence measures considered are equivalent for binary classification problems, in order to experimentally compare them we are going to discard all the databases having binary classes, thus keeping only 14 from the 31 original databases for the new experiments. As we aim to study all the pairwise comparisons between the five confidence measures, without using a control method, we must use a post-hoc test appropriate for this task; we have used the Shaffer test (García and Herrera, 2008).

We have not found almost any statistically significant differences among the confidence measures. To be exact, we only found one significant difference among all the 60 cases

considered ($\beta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and the six base classifiers): ϕ_d was significantly better than ϕ_e when using $F_{0.5}$ and the algorithm `IBk`. This may be due in part to the fewer databases being considered. In Tables 6 and 7 we display the number of times that each confidence measure was better than the others, when using F_β and EP_ρ, respectively.

Although we cannot speak of significant differences, we can clearly observe that the confidence measures based on difference of probabilities (ϕ_d), standard deviation (ϕ_s) and maximum probability (ϕ_m) tend to outperform those based on entropy (ϕ_e) and Euclidean distance (ϕ_{ed}).

4.2.4. Evaluating the oracle's effort

In order to evaluate how many queries are required to achieve the performance gain of the interactive classifiers, i.e. to evaluate the oracle's effort, in the previous experiments we have also computed the percentages of interaction of the classifier with the oracle, $100 * n_i/N$. In Tables 8 and 9 these percentages, averaged across the 31 databases, are displayed for each algorithm and five different values of the parameters β and ρ (in the range where interactive classification is beneficial). We have selected only one confidence measure, namely the one based on the difference of probabilities ϕ_d, the results for the other measures are very similar. Observe that the columns with label 0.5 in Tables 8 and 9 display the percentage of interactions corresponding to the values of $F_{0.5}$ and $EP_{0.5}$ in the column with label ϕ_d in Table 3.

Table 6
Number of times that the confidence measure in row i is better than the confidence measure in column j when using F_β, with $\beta = 0.1, 0.2, 0.3, 0.4, 0.5$ and the six base classifiers.

F_β	ϕ_e	ϕ_m	ϕ_d	ϕ_s	ϕ_{ed}	Total best
ϕ_e	–	12	14	6	13	45
ϕ_m	17	–	10	10	18	55
ϕ_d	16	19	–	15	19	69
ϕ_s	24	18	14	–	18	74
ϕ_{ed}	16	12	10	11	–	49
Total worst	73	61	48	42	68	

Table 7
Number of times that the confidence measure in row i is better than the confidence measure in column j when using EP_ρ, with $\rho = 0.1, 0.2, 0.3, 0.4, 0.5$ and the six base classifiers.

EP_ρ	ϕ_e	ϕ_m	ϕ_d	ϕ_s	ϕ_{ed}	Total best
ϕ_e	–	13	12	10	19	54
ϕ_m	17	–	14	15	18	64
ϕ_d	16	15	–	18	22	71
ϕ_s	19	12	12	–	20	63
ϕ_{ed}	11	11	7	9	–	38
Total worst	63	51	45	52	79	

Table 8
Percentages of interaction of the classifier with the oracle, when using ϕ_d and F_β for different values of β.

β	0.1	0.2	0.3	0.4	0.5
HC	43.67	31.94	23.62	18.54	12.93
IBk	40.51	34.61	27.09	21.56	15.62
J48	43.00	26.16	15.80	10.80	6.37
LOG	32.95	24.50	18.86	13.93	10.73
MP	22.14	16.80	14.26	12.45	10.83
NB	47.47	36.33	27.56	22.01	16.88

Table 9
Percentages of interaction of the classifier with the oracle, when using ϕ_d and EP_ρ for different values of ρ.

ρ	0.1	0.2	0.3	0.4	0.5
HC	34.49	22.17	15.53	8.78	3.63
IBk	34.27	24.02	15.30	8.81	4.52
J48	31.75	14.42	7.40	3.05	0.86
LOG	25.42	17.14	10.41	6.80	3.38
MP	18.33	13.26	11.01	8.57	6.37
NB	41.88	26.01	18.93	11.26	5.97

We can observe that the percentage of interaction increases considerably as we decrease the value of the parameters β and ρ, although the increment is quite dependent on the base algorithm being considered. It seems that `NB` is the algorithm that requires more interactions with the oracle, whereas `J48`, `LOG` and `MP` need less interactions (`MP` is better for smaller values of β and ρ, whereas `J48` performs better for greater values of the parameters).

In Fig. 2 we can take a closer look at the results (percentage of interaction) for the different databases obtained by `NB`, `MP` and `J48`, for the parameter $\beta = 0.5$. Apart from the variability due to the different algorithms, we can see that the results also depend heavily on the databases: there are differences greater than 35% (for `NB` and `MP`) and 19% (for `J48`) from a database to another.

4.2.5. Comparing with a random decision strategy

Another interesting question[5] is to what extent the decision strategies proposed, based on a confidence measure and a threshold, are better than a strategy which randomly asks the oracle a given percentage of predictions. For example, if a classifier was always wrong (for all the test examples, i.e. $n_c = 0$), a random decision strategy would be equivalent to the other strategies (provided we are using the F_β measure).

[5]Posed by a reviewer.

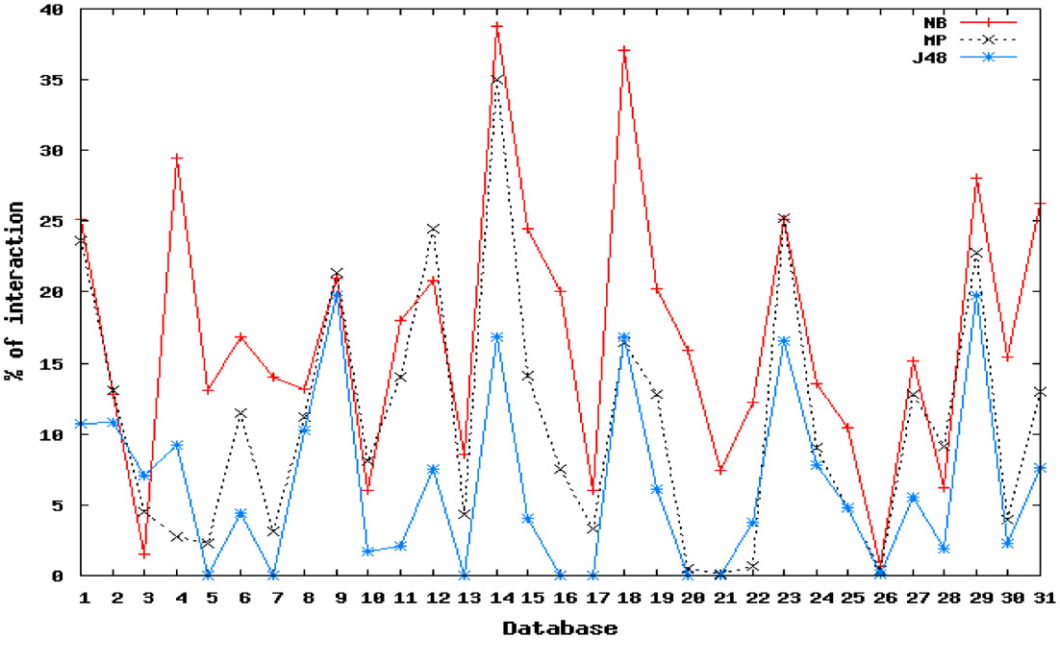

Fig. 2. Percentages of interaction obtained by NB, MP and J48 for each of the databases, when using ϕ_d and F_β with $\beta = 0.5$.

To this end we have carried out experiments with the following random decision strategy: first, we try to determine, using only the training set, which percentage of random queries would give the best result (the best value of our performance measure, either F_β or EP_ρ). To do that, for each case (each algorithm, each problem and each performance measure) we try different percentages $(5\%, 10\%, 15\%, \ldots, 90\%, 95\%)$ and perform random sampling using this percentage to select the examples to query for. We repeat the process several times (10 times) with each percentage and get the average performance, in order to obtain more stable and reliable results. Once we have determined the best percentage for each case, we apply random sampling using this percentage to the corresponding test set and compute the performance. We used the random sampling algorithm proposed by Knuth (1997).

The results of these experiments are displayed in Tables 10 and 11, for the performance measures F_β and EP_ρ, respectively. These tables show a summary of the results averaged across the 31 databases and the 6 algorithms, for the different strategies, including the random strategy as well as the averages of the best estimated percentage used by this strategy.

We can clearly observe that the random strategy performs poorly: it is always considerably worse than any of the informed strategies, and it is also worse (except in one case) than the non-interactive classifier. The results of this strategy are progressively worse as the values of the parameters β and ρ increase. The averages of the best percentages remain more or less stable around 20%. In order to take a closer look at the results of the random strategy, Table 12 displays them for the parameters $\beta = 0.5$ and $\rho = 0.5$, breakdown by algorithm, which can be

Table 10
Macro-average values of the performance measure F_β across the 31 databases and the 6 learning algorithms, for each confidence measure and different values of β.

β	ϕ_e	ϕ_m	ϕ_d	ϕ_s	ϕ_{ed}	Random	Best %
0.1	0.9301	0.9308	0.9309	0.9310	0.9306	0.8298	18.39
0.2	0.9176	0.9181	0.9181	0.9180	0.9178	0.8087	18.10
0.3	0.9026	0.9027	0.9029	0.9027	0.9028	0.7705	19.14
0.4	0.8903	0.8905	0.8905	0.8905	0.8903	0.7410	19.94
0.5	0.8813	0.8815	0.8814	0.8815	0.8811	0.6972	19.95
0.6	0.8749	0.8751	0.8749	0.8750	0.8746	0.6645	19.78
0.7	0.8707	0.8710	0.8706	0.8707	0.8704	0.6329	19.89
0.8	0.8686	0.8687	0.8686	0.8686	0.8686	0.6018	20.31
0.9	0.8670	0.8669	0.8669	0.8668	0.8668	0.5715	19.61
1.0	0.8664	0.8662	0.8663	0.8662	0.8662	0.5606	20.20

No interaction: 0.8672.

Table 11
Macro-average values of the performance measure EP_ρ across the 31 databases and the 6 learning algorithms, for each confidence measure and different values of ρ.

ρ	ϕ_e	ϕ_m	ϕ_d	ϕ_s	ϕ_{ed}	Random	Best%
0.1	0.9320	0.9325	0.9325	0.9325	0.9325	0.9048	42.40
0.2	0.9040	0.9040	0.9042	0.9040	0.9040	0.8368	21.27
0.3	0.8861	0.8862	0.8863	0.8862	0.8860	0.7841	18.97
0.4	0.8740	0.8742	0.8741	0.8742	0.8739	0.7315	19.37
0.5	0.8683	0.8685	0.8684	0.8683	0.8683	0.6779	19.13
0.6	0.8661	0.8661	0.8661	0.8660	0.8661	0.6269	19.52
0.7	0.8648	0.8649	0.8649	0.8648	0.8649	0.5735	19.80
0.8	0.8639	0.8639	0.8640	0.8638	0.8640	0.5217	20.00
0.9	0.8632	0.8632	0.8633	0.8631	0.8633	0.4699	19.52

No interaction: 0.8672.

Table 12
Macro-average values of the performance measures $F_{0.5}$ and $EP_{0.5}$ across the 31 databases, for each algorithm, using the random strategy. They can be compared with the results in Table 3.

Algorithm	$F_{0.5}$	$EP_{0.5}$
HC	0.6952	0.6760
IBk	0.6950	0.6825
J48	0.6916	0.6765
LOG	0.6998	0.6744
MP	0.7059	0.6813
NB	0.6955	0.6767

directly compared with those found in Table 3 for the other strategies. We can see that the behavior of the random strategy is uniformly bad across all the algorithms.

5. Concluding remarks

In this paper we have centered on a scenario which is not usually considered in the field of automatic classification: an already built classifier having the possibility of interacting with a human oracle, in such a way that the classifier must decide when to classify an instance and when it should query the correct label of this instance to the oracle. We have studied two fundamental problems arising in this scenario: how to measure the performance of such a system, taking into account both the accuracy of the classifier and the cost of the human intervention, and how to decide whether or not to query.

With respect to the first problem, after specifying some properties that any coherent evaluation measure for interactive classification should satisfy, and showing that several apparently reasonable measures are not coherent, we have proposed two coherent measures. One is based on the concepts of precision and recall, and the usual way of aggregating them through the F measure. The other is based of the idea of computing the average costs of using the interactive classifier, taking explicitly into account the costs of misclassifying and querying. Both measures depend on a single parameter which determines the contribution of each of the two components of each measure.

Next, we have studied strategies that an interactive classifier may use to interact with the human oracle. We have assumed that the output of the classifier is a posterior probability distribution over the possible categories and then a confidence measure based on this distribution and a threshold are used to define the decision rule: to ask the oracle if the confidence measure is below the threshold. In addition to several confidence measures used in the active learning field, we have proposed two additional measures based on the sample standard deviation and the Euclidean distance.

Finally, through an extensive experimental evaluation, using 6 standard classifiers and 31 databases, we have shown that: (1) an interactive classifier is significantly better than its non-interactive counterpart (with respect to the coherent evaluation measures defined), provided that an appropriate threshold for the decision strategy can be estimated (and we can do it) and the parameters of the two evaluation measures give more importance to precision and to the cost of misclassification; (2) some of the confidence measures considered (concretely those based on difference of probabilities, standard deviation and maximum probability) outperform the others (those based on entropy and Euclidean distance), although the differences are not statistically significant. In turn all of the confidence measures outperform an uninformed interactive strategy which randomly asks the oracle a given percentage of predictions.

For future research we would like to study whether some kind of external information (different from the own prediction suggested by the classifier for the given instance) could be used to improve the classifier's decision strategy. For example, information gathered during the training and testing of the base classifier. Another interesting task would be to study the influence of some characteristics of the problems (e.g. the number of classes, or the distribution of the class variable) on the behavior of the interactive classifier. We are also planning, for some specific base classifiers, to study and evaluate methods to re-use the information provided by the oracle in order to re-learn the interactive classifier and improve its performance. If this information would include not only the labels but also some arguments supporting the decisions (Mozina et al., 2007) then the quality of the re-learned model could be even greater.

Acknowledgments

This work has been jointly supported by the Spanish Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía (P09-TIC-4526), the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and the Spanish Ministerio de Ciencia e Innovación (TIN2011-28538-C02-02).

References

Aha, D., Kibler, D., 1991. Instance-based learning algorithms. Machine Learning 6, 37–66.

Blake, C.L., Merz, C.J., 1998. UCI Repository of Machine Learning Databases. University of California, Irvine, Department of Information and Computer Sciences 〈http://www.ics.uci.edu/~mlearn/MLRepository.html〉.

Cohn, D., Atlas, L., Ladner, R., 1994. Improving generalization with active learning. Machine Learning 15 (2), 201–221.

de Campos, L.M., Romero, A.E., 2009. Bayesian network models for hierarchical text classification from a thesaurus. International Journal of Approximate Reasoning 50 (7), 932–944.

Demsar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Dumais, S., Chen, H., 2000. Hierarchical classification of Web content. In: Proceedings of the 23rd Annual International ACM SIGIR Conference, pp. 256–263.

Fayyad, U.M., Irani, K.B., 1993. Multi-valued interval discretization of continuous-valued attributes for classification learning. In: Proceedings of

the Thirteenth International Joint Conference on Artificial Intelligence, pp. 1022–1027.

García, S., Herrera, F., 2008. An extension of "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. Journal of Machine Learning Research 9, 2677–2694.

Golub, K., 2006. Automated subject classification of textual web documents. Journal of Documentation 62 (3), 350–371.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA data mining software: an update. SIGKDD Explorations 11 (1), 10–18.

Haykin, S., 1998. Neural Networks: A Comprehensive Foundation, second ed. Prentice Hall.

Heckerman, D., Geiger, D., Chickering, D.M., 1995. Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning 20, 197–243.

Helmbold, D., Panizza, S., 1997. Some label efficient learning results. In: Proceedings of the Tenth Annual Conference on Computational Learning Theory, pp. 218–230.

Hilbe, J.M., 2009. Logistic Regression Models. Chapman & Hall/CRC Press.

Knuth, D.E., 1997. The Art of Computer Programming, Vol. 2. Addison-Wesley.

Kohavi, R., John, G.H., Long, R., Manley, D., Pfleger, K., 1994. MLC++: A machine learning library in C++. In: Proceedings of the Sixth International Conference on Tools with Artificial Intelligence, pp. 740–743.

Lewis, D., Gale, W., 1994. A sequential algorithm for training text classifiers. In: Proceedings of the ACM SIGIR Conference, pp. 3–12.

Littlestone, N., 1988. Learning quickly when irrelevant attributes abound: a new linear threshold algorithm. Machine Learning 2, 285–318.

Littlestone, N., Warmuth, M.K., 1994. The weighted majority algorithm. Information and Computation 108 (2), 212–261.

Mozina, M., Zabkar, J., Bratko, I., 2007. Argument based machine learning. Artificial Intelligence 171, 922–937.

Quinlan, R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.

Sebastiani, F., 2002. Machine Learning in automated text categorization. ACM Computing Surveys 34, 1–47.

Settles, B., 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Settles, B., Craven, M., 2008. An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1069–1078.

Stumpf, S., Rajaram, V., Li, L., Wong, W.K., Burnett, M., Dietterich, T., Sullivan, E., Herlocker, J., 2009. Interacting meaningfully with machine learning systems: three experiments. International Journal of Human-Computer Studies 67, 639–662.

van Rijsbergen, C.J., 1979. Information Retrieval, second ed. Butterworth.

Ware, M., Frank, E., Holmes, G., Hall, M., Witten, I.H., 2001. Interactive machine learning: letting users build classifiers. International Journal of Human-Computer Studies 55 (3), 281–292.