

# Cost-minimising strategies for data labelling : optimal stopping and active learning

Christos Dimitrakakis  
christos.dimitrakakis@gmail.com

Christian Savu-Krohn  
christian.savu-krohn@unileoben.ac.at

Chair of Information Technology, University of Leoben  
Leoben A-8700, Austria

November 15, 2007

## Abstract

Supervised learning deals with the inference of a distribution over an output or label space  $\mathcal{Y}$  conditioned on points in an observation space  $\mathcal{X}$ , given a training dataset  $D$  of pairs in  $\mathcal{X} \times \mathcal{Y}$ . However, in a lot of applications of interest, acquisition of large amounts of observations is easy, while the process of generating labels is time-consuming or costly. One way to deal with this problem is *active* learning, where points to be labelled are selected with the aim of creating a model with better performance than that of an model trained on an equal number of randomly sampled points. In this paper, we instead propose to deal with the labelling cost directly: The learning goal is defined as the minimisation of a cost which is a function of the expected model performance and the total cost of the labels used. This allows the development of general strategies and specific algorithms for (a) optimal stopping, where the expected cost dictates whether label acquisition should continue (b) empirical evaluation, where the cost is used as a performance metric for a given combination of inference, stopping and sampling methods. Though the main focus of the paper is optimal stopping, we also aim to provide the background for further developments and discussion in the related field of active learning.

## 1 Introduction

Much of classical machine learning deals with the case where we wish to learn a target concept in the form of a function  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , when all we have is a finite set of examples  $D = \{(x_i, y_i)\}_{i=1}^n$ . However, in many practical settings, it turns out that for each example  $i$  in the set only the observations  $x_i$  are available, while

the availability of observations  $y_i$  is restricted in the sense that either (a) they are only observable for a subset of the examples (b) further observations may only be acquired at a cost. In this paper we deal with the second case, where we can actually obtain labels for any  $i \in D$ , but doing so incurs a cost. Active learning algorithms (i.e. [1, 2]) deal indirectly with this by selecting examples which are expected to increase accuracy the most. However, the basic question of whether new examples should be queried at all is seldom addressed.

This paper deals with the labelling cost explicitly. We introduce a cost function that represents the trade-off between final performance (in terms of generalisation error) and querying costs (in terms of the number of labels queried). This is used in two ways. Firstly, as the basis for creating cost-dependent stopping rules. Secondly, as the basis of a comparison metric for learning algorithms and associated stopping algorithms.

To expound further, we decide when to stop by estimating the expected performance gain from querying additional examples and comparing it with the cost of acquiring more labels. One of the main contributions is the development of methods for achieving this in a Bayesian framework. While due to the nature of the problem there is potential for misspecification, we nevertheless show experimentally that the stopping times we obtain are close to the optimal stopping times.

We also use the trade-off in order to address the lack of a principled method for comparing different active learning algorithms under conditions similar to real-world usage. For such a comparison a method for choosing stopping times independently of the test set is needed. Combining stopping rules with active learning algorithms allows us to objectively compare active learning algorithms for a range of different labelling costs.

The paper is organised as follows. Section 1.1 introduces the proposed cost function for when labels are costly, while Section 1.2 discusses related work. Section 2 derives a Bayesian stopping method that utilises the proposed cost function. Some experimental results illustrating the proposed evaluation methodology and demonstrating the use of the introduced stopping method are presented in Section 3. The proposed methods are not flawless, however. For example, the algorithm-independent stopping rule requires the use of i.i.d. examples, which may interfere with its coupling to an active learning algorithm. We conclude with a discussion on the applicability, merits and deficiencies of the proposed approach to optimal stopping and of principled testing for active learning.

## 1.1 Combining Classification Error and Labelling Cost

There are many applications where raw data is plentiful, but labelling is time consuming or expensive. Classic examples are speech and image recognition, where it is easy to acquire hours of recordings, but for which transcription and labelling are laborious and costly. For this reason, we are interested in querying labels from a given dataset such that we find the optimal balance between the cost of labelling and the classification error of the hypothesis inferred from the labelled examples. This arises naturally from the following cost function.

Let some algorithm  $F$  which queries labels for data from some unlabelled dataset  $D$ , incurring a cost  $\gamma \in [0, \infty)$  for each query. If the algorithm stops after querying labels of examples  $d_1, d_2, \dots, d_t$ , with  $d_i \in [1, |D|]$ . it will suffer a total cost of  $\gamma t$ , plus a cost depending on the generalisation error. Let  $f(t)$  be the hypothesis obtained after having observed  $t$  examples and corresponding to the generalisation error  $\mathbf{E}[R|f(t)]$  be the generalisation error of the hypothesis. Then, we define the total cost for this specific hypothesis as

$$\mathbf{E}[C_\gamma|f(t)] = \mathbf{E}[R|f(t)] + \gamma t. \quad (1)$$

We may use this cost as a way to compare learning and stopping algorithms, by calculating the expectation of  $C_\gamma$  conditioned on different algorithm combinations, rather than on a specific hypothesis.

In addition, this cost function can serve as a formal framework for active learning. Given a particular dataset  $D$ , the optimal subset of examples to be used for training will be  $D^* = \operatorname{argmin}_i \mathbf{E}(R|F, D_i) + \gamma|D_i|$ . The ideal, but unrealisable, active learner in this framework would just use labels of the subset  $D^*$  for training.

Thus, these notions of optimality can in principle be used both for deriving stopping and sampling algorithms and for comparing them. Suitable metrics of expected real-world performance will be discussed in the next section. Stopping methods will be described in Section 2.

## 1.2 Related Work

In the active learning literature, the notion of an objective function for trading off classification error and labelling cost has not yet been adopted. However, a number of both qualitative and quantitative metrics were proposed in order to compare active learning algorithms. Some of the latter are defined as summary statistics over some subset  $\mathcal{T}$  of the possible stopping times. This is problematic as it could easily be the case that there exists  $\mathcal{T}_1, \mathcal{T}_2$  with  $\mathcal{T}_1 \subset \mathcal{T}_2$ , such that when comparing algorithms over  $\mathcal{T}_1$  we get a different result than when we are comparing them over a larger set  $\mathcal{T}_2$ . Thus, such measures are not easy to interpret since the choice of  $\mathcal{T}$  remains essentially arbitrary. Two examples are (a) the *percentage reduction in error*, where the percentage reduction in error of one algorithm over another is averaged over the whole learning curve [3, 4] and (b) the average number of times one algorithm is significantly better than the other during an arbitrary initial number of queries, which was used in [5]. Another metric is the *data utilisation ratio* used in [5, 4, 6], which is the amount of data required to reach some specific error rate. Note that the selection of the appropriate error rate is essentially arbitrary; in both cases the concept of the *target error rate* is utilised, which is the average test error when almost all the training set has been used.

Our setting is more straightforward, since we can use (1) as the basis for a performance measure. Note that we are not strictly interested in comparing hypotheses  $f$ , but algorithms  $F$ . In particular, we can calculate the expected

cost given a learning algorithm  $F$  and an associated stopping algorithm  $Q_F(\gamma)$ , which is used to select the *stopping time*  $T$ . From this follows that the expected cost of  $F$  when coupled with  $Q_F(\gamma)$  is

$$v_e(\gamma, F, Q_F) \equiv \mathbf{E}[C_\gamma | F, Q_F(\gamma)] = \sum_t (\mathbf{E}[R|f(t)] + \gamma t) \mathbf{P}[T = t | F, Q_F(\gamma)] \quad (2)$$

By keeping one of the algorithms fixed, we can vary the other in order to obtain objective estimates of their performance difference. In addition, we may want to calculate the expected performance of algorithms for a range of values of  $\gamma$ , rather than a single value, in a manner similar to what [7] proposed as an alternative to ROC curves. This will require a stopping method  $Q_F(\gamma)$  which will ideally stop querying at a point that minimises  $\mathbf{E}(C_\gamma)$ .

The stopping problem is not usually mentioned in the active learning literature and there are only a few cases where it is explicitly considered. One such case is [2], where it is suggested to stop querying when no example lies within the SVM margin. The method is used indirectly in [8], where if this event occurs the algorithm tests the current hypothesis<sup>1</sup>, queries labels for a new set of unlabelled examples<sup>2</sup> and finally stops if the error measured there is below a given threshold; similarly, [9] introduced a bounds-based stopping criterion that relies on an allowed error rate. These are reasonable methods, but there exists no formal way of incorporating the cost function considered here within them. For our purpose we need to calculate the expected reduction in classification error when querying new examples and compare it with the labelling cost. This fits nicely within the statistical framework of optimal stopping problems.

## 2 Stopping Algorithms

An optimal stopping problem under uncertainty is generally formulated as follows. At each point in time  $t$ , the experimenter needs to make a decision  $a \in A$ , for which there is a *loss function*  $\mathcal{L}(a|w)$  defined for all  $w \in \Omega$ , where  $\Omega$  is the set of all possible universes. The experimenter’s uncertainty about which  $w \in \Omega$  is true is expressed via the distribution  $\mathbf{P}(w|\xi_t)$ , where  $\xi_t$  represents his belief at time  $t$ . The *Bayes risk* of taking an action at time  $t$  can then be written as  $\rho_0(\xi_t) = \min_a \sum_w \mathcal{L}(a, w) \mathbf{P}(w|\xi_t)$ . Now, consider that instead of making an immediate decision, he has the opportunity to take  $k$  more observations  $D_k$  from a sample space  $S^k$ , at a cost of  $\gamma$  per observation, thus allowing him to update his belief to  $\mathbf{P}(w|\xi_{t+k}) \equiv \mathbf{P}(w|D_k, \xi_t)$ . What the experimenter must do in order to choose between immediately making a decision  $a$  and continuing sampling, is to compare the risk of making a decision now with the cost of making  $k$  observations plus the risk of making a decision after  $k$  timesteps, when the extra data would enable a more informed choice. In other words, one should stop and

<sup>1</sup>i.e. a classifier for a classification task

<sup>2</sup>Though this is not really an i.i.d. sample from the original distribution except when  $|D|-t$  is large.

make an immediate decision if the following holds for all  $k$ :

$$\rho_0(\xi_t) \leq \gamma k + \int_{S^k} p(D_k = s | \xi_t) \min_a \left[ \sum_w \mathcal{L}(a, w) \mathbf{P}(w | D_k = s, \xi_t) \right] ds. \quad (3)$$

We can use the same formalism in our setting. In one respect, the problem is simpler, as the only decision to be made is when to stop and then we just use the currently obtained hypothesis. The difficulty lies in estimating the expected error. Unfortunately, the metrics used in active learning methods for selecting new examples (see [5] for a review) do not generally include calculations of the expected performance gain due to querying additional examples.

There are two possibilities for estimating this performance gain. The first is an algorithm-independent method, described in detail in Sec. 2.1, which uses a set of convergence curves, arising from theoretical convergence properties. We employ a Bayesian framework to infer the probability of each convergence curve through observations of the error on the next randomly chosen example to be labelled. The second method, outlined in Sec. 4, relies upon a classifier with a probabilistic expression of its uncertainty about the class of unlabelled examples, but is much more computationally expensive.

## 2.1 When no Model is Perfect: Bayesian Model Selection

The presented Bayesian formalism for optimal sequential decisions follows [10]. We require maintaining a belief  $\xi_t$  in the form of a probability distribution over the set of possible universes  $w \in \Omega$ . Furthermore, we require the existence of a well-defined cost for each  $w$ . Then we can write the Bayes risk as in the left side of (3), but ignoring the minimisation over  $A$  as there is only one possible decision to be made after stopping,

$$\rho_0(\xi_t) = \mathbf{E}(R_t | \xi_t) = \sum_{w \in \Omega} \mathbf{E}(R_t | w) \mathbf{P}(w | \xi_t), \quad (4)$$

which can be extended to continuous measures without difficulty. We will write the expected risk according to our belief at time  $t$  for the optimal procedure taking at most  $k$  more samples as

$$\rho_{k+1}(\xi_t) = \min \{ \rho_0(\xi_t), \mathbf{E}[\rho_k(\xi_{t+1}) | \xi_t] + \gamma \}. \quad (5)$$

This implies that at any point in time  $t$ , we should ignore the cost for the  $t$  samples we have paid for and are only interested in whether we should take additional samples. The general form of the stopping algorithm is defined in Alg. 1. Note that the horizon  $K$  is a necessary restriction for computability. A larger value of  $K$  leads to potentially better decisions, as when  $K \rightarrow \infty$ , the bounded horizon optimal decision approaches that of the optimal decision in the unbounded horizon setting, as shown for example in Chapter 12 of [10]. Even with finite  $K > 1$ , however, the computational complexity is considerable, since we will have to additionally keep track of how our future beliefs  $\mathbf{P}(w | \xi_{t+k})$  will evolve for all  $k \leq K$ .

---

**Algorithm 1** A general bounded stopping algorithm using Bayesian inference.

---

Given a dataset  $D$  and any learning algorithm  $F$ , an initial belief  $\mathbf{P}(w \mid \xi_0)$  and a method for updating it, and additionally a known query cost  $\gamma$ , and a horizon  $K$ ,

- 1: **for**  $t = 1, 2, \dots$  **do**
  - 2:   Use  $F$  to query a new example  $i \in D$  and obtain  $f(t)$ .
  - 3:   Observe the empirical error estimate  $v_t$  for  $f(t)$ .
  - 4:   Calculate  $\mathbf{P}(w \mid \xi_t) = \mathbf{P}(w \mid v_t, \xi_{t-1})$
  - 5:   **if**  $\nexists k \in [1, K] : \rho_k(\xi_t) < \rho_0(\xi_t)$  **then**
  - 6:     Exit.
  - 7:   **end if**
  - 8: **end for**
- 

## 2.2 The OBSV Algorithm

In this paper we consider a specific one-step bounded stopping algorithm that uses independent validation examples for observing the empirical error estimate  $r_t$ , which we dub OBSV and is shown in detail in Alg. 2. The algorithm considers hypotheses  $w \in \Omega$  which model how the generalisation error  $r_t$  of the learning algorithm changes with time. We assume that the initial error is  $r_0$  and that the algorithm always converges to some unknown  $r_\infty \equiv \lim_{t \rightarrow \infty} r_t$ . Furthermore, we need some observations  $v_t$  that will allow us to update our beliefs over  $\Omega$ . The remainder of this section discusses the algorithm in more detail.

### 2.2.1 Steps 1-5, 11-12. Initialisation and Observations

We begin by splitting the training set  $D$  in two parts:  $D_A$ , which will be sampled without replacement by the *active learning algorithm* (if there is one) and  $D_R$ , which will be *uniformly* sampled without replacement. This condition is necessary in order to obtain i.i.d. samples for the inference procedure outlined in the next section. However, if we only sample randomly, and we are not using an active learning algorithm then we do not need to split the data and we can set  $D_A = \emptyset$ .

At each timestep  $t$ , we will use a sample from  $D_R$  to update  $p(w)$ . If we then expect to reduce our future error sufficiently, we will query an example from  $D_A$  using  $F$  and subsequently update the classifier  $f$  with both examples. Thus, not only are the observations used for inference independent and identically distributed, but we are also able to use them to update the classifier  $f$ .

### 2.2.2 Step 6. Updating the Belief

We model the learning algorithm as a process which asymptotically converges from an initial error  $r_0$  to the unknown final error  $r_\infty$ . Each model  $w$  will be a *convergence estimate*, a model of how the error converges from the initial to the final error rate. More precisely, each  $w$  corresponds to a function  $h_w : \mathbb{N} \rightarrow [0, 1]$

that models how close we are to convergence at time  $t$ . The predicted error at time  $t$  according to  $w$ , given the initial error  $r_0$  and the final error  $r_\infty$ , will be

$$g_w(t | r_0, r_\infty) = r_0 h_w(t) + r_\infty [1 - h_w(t)]. \quad (6)$$

We find it reasonable to assume that  $p(w, r_0, r_\infty) = p(w)p(r_0)p(r_\infty)$ , i.e. that the convergence rates do not depend upon the initial and final errors.

We may now use these predictions together with some observations to update  $p(w, r_\infty | \xi)$ . More specifically, if  $\mathbf{P}[r_t = g_w(t | r_0, r_\infty) | r_0, r_\infty, w] = 1$  and we take  $m_t$  independent observations  $\mathbf{z}_t = (z_t(1), z_t(2), \dots, z_t(m_t))$  of the error with mean  $v_t$ , the likelihood will be given by the Bernoulli density

$$p(\mathbf{z}_t | w, r_0, r_\infty) = (g_w(t | r_0, r_\infty))^{v_t} [1 - g_w(t | r_0, r_\infty)]^{1-v_t} {}^{m_t}. \quad (7)$$

Then it is simple to obtain a posterior density for both  $w$  and  $r_\infty$ ,

$$p(w | \mathbf{z}_t) = \frac{p(w)}{p(\mathbf{z}_t)} \int_0^1 p(\mathbf{z}_t | w, r_0, r_\infty = u) p(r_\infty = u | w) du \quad (8a)$$

$$p(r_\infty | \mathbf{z}_t) = \frac{p(r_\infty)}{p(\mathbf{z}_t)} \int_\Omega p(\mathbf{z}_t | w, r_0, r_\infty) p(w | r_\infty) dw. \quad (8b)$$

Starting with a prior distribution  $p(w | \xi_0)$  and  $p(r_\infty | \xi_0)$ , we may sequentially update our belief using (8) as follows:

$$p(w | \xi_{t+1}) \equiv p(w | \mathbf{z}_t, \xi_t) \quad (9a)$$

$$p(r_\infty | \xi_{t+1}) \equiv p(r_\infty | \mathbf{z}_t, \xi_t). \quad (9b)$$

The realised convergence for a particular training data set may differ substantially from the expected convergence: the average convergence curve will be smooth, while any specific instantiation of it will not be. More formally, the *realised error* given a specific training dataset is  $q_t \equiv \mathbf{E}[R_t | D^t]$ , where  $D^t \sim \mathcal{D}^t$ , while the *expected error* given the data distribution is  $r_t \equiv \mathbf{E}[R_t] = \int_{S^t} \mathbf{E}[R_t | D^t] \mathbf{P}(D_t) dD_t$ . The smooth convergence curves that we model would then correspond to models for  $r_t$ .

Fortunately, in our case we can estimate a distribution over  $r_t$  without having to also estimate a distribution for  $q_t$ , as this is integrated out for observations  $z \in \{0, 1\}$

$$p(z | q_t) = q_t^z (1 - q_t)^{1-z} \quad (10a)$$

$$p(z | r_t) = \int_0^1 p(z | q_t) p(q_t = u | r_t) du = r_t^z (1 - r_t)^{1-z}. \quad (10b)$$

### 2.2.3 Step 5. Deciding whether to Stop

We may now use the distribution over the models to predict the error should we choose to add  $k$  more examples. This is simply

$$\mathbf{E}[R_{t+k} | \xi_t] = \int_0^1 \int_\Omega g_w(t+k | r_0, r_\infty) p(w | \xi_t) p(r_\infty | \xi_t) dw dr_\infty.$$

The calculation required for step 8 of OBSV follows trivially.

---

**Algorithm 2** OBSV, a specific instantiation of the bounded stopping algorithm.

Given a dataset  $D$  with examples in  $N_c$  classes and any learning algorithm  $F$ , initial beliefs  $\mathbf{P}(w \mid \xi_0)$  and  $\mathbf{P}(r_\infty \mid \xi_0)$  and a method for updating them, and additionally a known query cost  $\gamma$  for discovering the class label  $y_i \in [1, \dots, n]$  of example  $i \in D$ ,

- 1: Split  $D$  into  $D_A, D_R$ .
  - 2:  $r_0 = 1 - 1/N_c$ .
  - 3: Initialise the classifier  $f$ .
  - 4: **for**  $t = 1, 2, \dots$  **do**
  - 5:   Sample  $i \in D_R$  without replacement and observe  $f(x_i), y_i$  to calculate  $v_t$ .
  - 6:   Calculate  $\mathbf{P}(w, r_\infty \mid \xi_t) \equiv \mathbf{P}(w, r_\infty \mid v_t, \xi_{t-1})$ .
  - 7:   If  $D_A \neq \emptyset$ , set  $k = 2$ , otherwise  $k = 1$ .
  - 8:   **if**  $\mathbf{E}[R_{t+k} \mid \xi_t] + k\gamma < \mathbf{E}[R_t \mid \xi_t]$  **then**
  - 9:     Exit.
  - 10:   **end if**
  - 11:   If  $D_A \neq \emptyset$ , use  $F$  to query a new example  $j \in D_A$  without replacement,  $D_T \leftarrow D_T \cup j$ .
  - 12:    $D_T \leftarrow D_T \cup i, f \leftarrow F(D_T)$ .
  - 13: **end for**
- 

#### 2.2.4 Specifics of the Model

What remains unspecified is the set of convergence curves that will be employed. We shall make use of curves related to common theoretical convergence results. It is worthwhile to keep in mind that we simply aim to find the combination of the available estimates that gives the best predictions. While none of the estimates might be particularly accurate, we expect to obtain reasonable stopping times when they are optimally combined in the manner described in the previous section. Ultimately, we expect to end up with a fairly narrow distribution over the possible convergence curves.

One of the weakest convergence results [11] is for sample complexity of order  $\mathcal{O}(1/\epsilon_t^2)$ , which corresponds to the convergence curve

$$h_q(t) = \sqrt{\frac{\kappa}{t + \kappa}}, \quad \kappa \geq 1 \quad (11)$$

Another common type is for sample complexity of order  $\mathcal{O}(1/\epsilon_t)$ , which corresponds to the curve

$$h_g(t) = \frac{\lambda}{t + \lambda}, \quad \lambda \geq 1 \quad (12)$$

A final possibility is that the error decreases exponentially fast. This is theoretically possible in some cases, as was proven in [9]. The resulting sample complexity of order  $\mathcal{O}(\log(1/\epsilon_t))$  corresponds to the convergence curve

$$h_{exp}(t) = \beta^t, \quad \beta \in (0, 1). \quad (13)$$

Since we do not know what appropriate values of the constants  $\beta$ ,  $\lambda$  and  $\kappa$ , are, we will model this uncertainty as an additional distribution over them, i.e.  $p(\beta | \xi_t)$ . This would be updated together with the rest of our belief distribution and could be done in some cases analytically. In this paper however we consider approximating the continuous densities by a sufficiently large set of models, one for each possible value of the unknown constants.

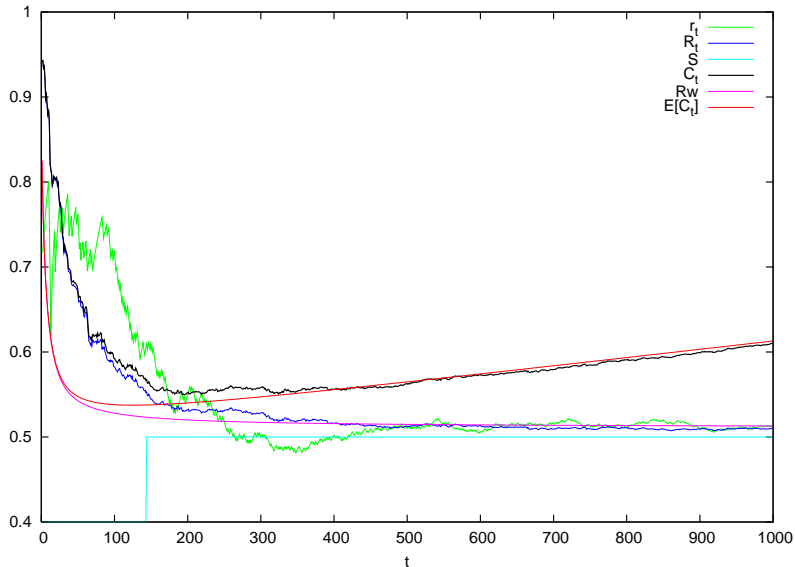


Figure 1: Illustration of the estimated error on a 10-class problem with a cost per label of  $\gamma = 0.001$ . On the vertical axis,  $r_t$  is the **history** of the predicted generalisation error, i.e.  $\mathbf{E}[r_t | \xi_{t-1}]$ , while  $R_t$  is the **generalisation error** measured on a test-set of size 10,000 and  $C_t$  is the corresponding **actual cost**. Finally,  $R_w$  and  $E[C_t]$  are the final **estimated** convergence and cost *curves* given all the observations. The stopping time is indicated by  $S$ , which equals 0.5 whenever Alg. 2 decides to stop and  $t$  is the number of iterations.

As a simple illustration, we examined the performance of the estimation and the stopping criterion in a simple classification problem with data of 10 classes, each with an equivariant Gaussian distribution in an 8-dimensional space. Each unknown point was simply classified as having the label closest to the empirical mean of the observations for each class. Examples were always chosen randomly.

As can be seen in Fig. 1, at the initial stages the estimates are inaccurate. This is because of two reasons: (a) The distribution over convergence rates is initially dominated by the prior. As more data is accumulated, there is better evidence for what the final error will be. (b) As we mentioned in the discussion of step 6, the realised convergence curve is much more random than the expected convergence curve which is actually modelled. However, as the number

of examples approaches infinity, the expected and realised errors converge. The stopping time for Alg. 2 (indicated by  $S$ ) is nevertheless relatively close to the optimal stopping time, as  $C_t$  appears to be minimised near 200. The following section presents a more extensive evaluation of this stopping algorithm.

### 3 Experimental Evaluation

The main purpose of this section is to evaluate the performance of the OBSV stopping algorithm. This is done by examining its cost and stopping time when compared to the optimal stopping time. Another aim of the experimental evaluation was to see whether mixed sampling strategies have an advantage compared to random sampling strategies with respect to the cost, when the stopping time is decided using a stopping algorithm that takes into account the labelling cost. Following [7], we plot performance curves for a range of values of  $\gamma$ , utilising multiple runs of cross-validation in order to assess the sensitivity of the results to the data. For each run, we split the data into a training set  $D$  and test set  $D_E$ , the training set itself being split into random and mixed sampling sets whenever appropriate.

More specifically, we compare the OBSV algorithm with the **oracle** stopping time. The latter is defined simply as the stopping time minimising the cost as this is measured on the independent test set for that particular run. We also compare **random** sampling with **mixed** sampling. In random sampling, we simply query unlabelled examples without replacement. For the mixed sampling procedure, we actively query an additional label for the example from  $D_A$  closest to the decision boundary of the current classifier, also without replacement. This strategy relies on the assumption that those labels are most informative [6], [4], [5] and thus convergence will be faster. Stopping times and cost ratio curves are shown for a set of  $\gamma$  values, for costs as defined in (2). These values of  $\gamma$  are also used as input to the stopping algorithm. The ratios are used both to compare stopping algorithms (OBSV versus the oracle) and sampling strategies (random sampling, where  $D_A = \emptyset$ , and mixed sampling, with  $|D_A| = |D_R|$ ). Average test error curves are also plotted for reference.

For the experiments we used two data sets from the UCI repository<sup>3</sup>: the Wisconsin breast cancer data set (**wdbc**) with 569 examples and the spambase database (**spam**) with 4601 examples. We evaluated **wdbc** and **spam** using 5 and 3 randomised runs of 3-fold stratified cross-validation respectively. The classifier used was AdaBoost [12] with 100 decision stumps as base hypotheses. Hence we obtain a total of 15 runs for **wdbc** and 9 for **spam**. We ran experiments for values of  $\gamma \in \{9 \cdot 10^{-k}, 8 \cdot 10^{-k}, \dots, 1 \cdot 10^{-k}\}$ , with  $k = 1, \dots, 7$ , and  $\gamma = 0$ . For every algorithm and each value of  $\gamma$  we obtain a different stopping time  $t_\gamma$  for each run. We then calculate  $v_e(\gamma, F, t_\gamma)$  as given in (2) on the corresponding test set of the run. By examining the averages and extreme values over all runs we are able to estimate the sensitivity of the results to the data.

<sup>3</sup><http://mllearn.ics.uci.edu/MLRepository.html>

The results comparing the oracle with OBSV for the **random** sampling strategy<sup>4</sup> are shown in Fig. 2. In Fig. 2(a), 2(b) it can be seen that the stopping times of OBSV and the oracle increase at a similar rate. However, although OBSV is reasonably close, on average it regularly stops earlier. This may be due to a number of reasons. For example, due to the prior, OBSV stops immediately when  $\gamma > 3 \cdot 10^{-2}$ . At the other extreme, when  $\gamma \rightarrow 0$  the cost becomes the test error and therefore the oracle always stops at latest at the minimum test error<sup>5</sup>. This is due to the stochastic nature of the realised error curve, which cannot be modelled; there, the perfect information that the oracle enjoys accounts for most of the performance difference. As shown in Fig. 2(c), 2(d), the extra cost induced by using OBSV instead of the oracle is bounded from above for most of the runs by factors of 2 to 5 for **wdbc** and around 0.5 for **spam**. The rather higher difference on **wdbc** is partially a result of the small dataset. Since we can only measure an error in quanta of  $1/|D_E|$ , any actual performance gain lower than this will be unobservable. This explains why the number of examples queried by the oracle becomes constant for a value of  $\gamma$  smaller than this threshold. Finally, this fact also partially explains the greater variation of the oracle’s stopping time in the smaller dataset. We expect that with larger test sets, the oracle’s behaviour would be smoother.

The corresponding comparison for the **mixed** sampling strategies is shown in Fig. 4(a), 4(b). We again observe the stopping times to increase at a similar rate, and OBSV to stop earlier on average than the oracle for most values of  $\gamma$  (Fig. 3(a), 3(b)). Note that the oracle selects the minimum test error at around 180 labels from **wdbc** and 1300 labels from **spam**, which for both data sets is only about a half of the number of labels the random strategy needs. OBSV tracks these stopping times closely. Over all, the fact that in both mixed and random sampling, the stopping times of OBSV and the oracle are usually well within the extreme value ranges, indicates a satisfactory performance.

Finally we compare the two sampling strategies directly as shown in Fig. 4, using the practical OBSV algorithm. As one might expect from the fact that the mixed strategy converges faster to a low error level, OBSV stops earlier or around the same time using the mixed strategy than it does for the random (Fig. 4(c), 4(d)). Those two facts together indicate that OBSV works as intended, since it stops earlier when convergence is faster. The results also show that when using OBSV as a stopping criterion mixed sampling is equal to or better than random sampling [Fig. 4(e), 4(f)]. However the differences are mostly not very significant.

## 4 Discussion

This paper discussed the interplay between a well-defined cost function, stopping algorithms and objective evaluation criteria and their relation to active learning. Specifically, we have argued that (a) learning when labels are costly is essentially

<sup>4</sup>The corresponding average test errors can be seen in Fig. 4(a), 4(b).

<sup>5</sup>This is obtained after about 260 labels on **wdbc** and 2400 labels on **spam**

a stopping problem (b) it is possible to use optimal stopping procedures based on a suitable cost function (c) the goal of active learning algorithms could also be represented by this cost function, (d) metrics on this cost function should be used to evaluate performance and finally that, (e) the stopping problem cannot be separately considered from either the cost function or the evaluation. To our current knowledge, these issues have not yet been sufficiently addressed.

For this reason, we have proposed a suitable cost function and presented a practical stopping algorithm which aims to be optimal with respect to this cost. Experiments with this algorithm for a specific prior show that it suffers only small loss compared to the optimal stopping time and is certainly a step forward from ad-hoc stopping rules.

On the other hand, while the presented stopping algorithm is an adequate first step, its combination with active learning is not perfectly straightforward since the balance between active and uniform sampling is a hyperparameter which is not obvious how to set.<sup>6</sup> An alternative is to use model-specific stopping methods. This could be done if we restrict ourselves to probabilistic classifiers, as for example in [1]; in this way we may be able to simultaneously perform optimal example selection and stopping. If such a classifier is not available for the problem at hand, then judicious use of frequentist techniques such as bootstrapping [13] may provide a sufficiently good alternative for estimating probabilities. Such an approach was advocated by [14] in order to optimally select examples; however in our case we could extend this to optimal stopping. Briefly, this can be done as follows. Let our belief at time  $t$  be  $\xi_t$ , such that for any point  $x \in \mathcal{X}$ , we have a distribution over  $\mathcal{Y}$ ,  $\mathbf{P}(y | x, \xi_t)$ . We may now calculate this over the whole dataset to estimate the realised generalisation error as the *expected error given the empirical data distribution and our classifier*

$$\mathbf{E}_D(v_t | \xi_t) = \frac{1}{|D|} \sum_{i \in D} [1 - \arg \max_y \mathbf{P}(y_i = y | x_i, \xi_t)]. \quad (14)$$

We can now calculate (14) for each one of the different possible labels. So we calculate the *expected error on the empirical data distribution if we create a new classifier from  $\xi_t$  by adding example  $i$*  as

$$\mathbf{E}_D(v_t | x_i, \xi_t) = \sum_{y \in \mathcal{Y}} \mathbf{P}(y_i = y | x_i, \xi_t) \mathbf{E}_D(v_t | x_i, y_i = y, \xi_t) \quad (15)$$

Note that  $\mathbf{P}(y_i = y | x_i, \xi_t)$  is just the probability of example  $i$  having label  $y$  according to our current belief,  $\xi_t$ . Furthermore,  $\mathbf{E}_D(v_t | x_i, y_i = y, \xi_t)$  results from calculating (14) using the classifier resulting from  $\xi_t$  and the added example  $i$  with label  $y$ . Then  $\mathbf{E}_D(v_t, \xi_t) - \mathbf{E}_D(v_t | x_i, \xi_t)$  will be the expected gain from using  $i$  to train. The (subjectively) optimal 1-step stopping algorithm is as follows: Let  $i^* = \arg \min_i \mathbf{E}_D(v_t | x_i, \xi_t)$ . Stop if  $\mathbf{E}_D(v_t | \xi_t) - \mathbf{E}_D(v_t | x_{i^*}, \xi_t) < \gamma$ .

A particular difficulty in the presented framework, and to some extent also in the field of active learning, is the choice of hyperparameters for the classifiers

---

<sup>6</sup>In this paper, the active and uniform sampling rates were equal.

themselves. For Bayesian models it is possible to select those that maximise the marginal likelihood.<sup>7</sup> One could alternatively maintain a set of models with different hyperparameter choices and separate convergence estimates. In that case, training would stop when there were no models for which the expected gain was larger than the cost of acquiring another label. Even this strategy, however, is problematic in the active learning framework, where each model may choose to query a different example's label. Thus, the question of hyperparameter selection remains open and we hope to address it in future work.

On another note, we hope that the presented exposition will at the very least increase awareness of optimal stopping and evaluation issues in the active learning community, lead to commonly agreed standards for the evaluation of active learning algorithms, or even encourage the development of example selection methods incorporating the notions of optimality suggested in this paper. Perhaps the most interesting result for active learning practitioners is the very narrow advantage of mixed sampling when a realistic algorithm is used for the stopping times. While this might only have been an artifact of the particular combinations of stopping and sampling algorithm and the datasets used, we believe that it is a matter which should be given some further attention.

### Acknowledgements

We would like to thank Peter Auer for helpful discussions, suggestions and corrections. This work was supported by the FSP/JRP Cognitive Vision of the Austrian Science Funds (FWF, Project number S9104-N13). This work was also supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

### References

- [1] Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. In Tesauro, G., Touretzky, D., Leen, T., eds.: *Advances in Neural Information Processing Systems*. Volume 7., The MIT Press (1995) 705–712
- [2] Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: *Proceedings of the 17th International Conference on Machine Learning, ICML-2000*. (2000) 839–846
- [3] Saar-Tsechansky, M., Provost, F.: Active learning for class probability estimation and ranking. In: *Proceedings of the 17th international joint conference on artificial intelligence - IJCAI 2001*. (2001) 911–920
- [4] Melville, P., Mooney, R.J.: Diverse ensembles for active learning. In: *Proceedings of the 21st International Conference on Machine Learning, ICML-2004*. (2004) 584–591

---

<sup>7</sup>Other approaches require the use of techniques such as cross-validation, which creates further complications.

- [5] Körner, C., Wrobel, S.: Multi-class ensemble-based active learning. In: Proceedings of ECML-2006. Volume 4212 of LNAI. (2006) 687–694
- [6] Abe, N., Mamitsuka, H.: Query learning strategies using boosting and bagging. In: ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1998) 1–9
- [7] Bengio, S., Mariéthoz, J., Keller, M.: The expected performance curve. In: International Conference on Machine Learning ICML'07. (2005)
- [8] Campbell, C., Cristianini, N., Smola, A.: Query learning with large margin classifiers. In: Proceedings of the 17th International Conference on Machine Learning, ICML-2000. (2000) 111–118
- [9] Balcan, M.F., Beygelzimer, A., Langford, J.: Agnostic active learning. In Cohen, W.W., Moore, A., eds.: 23rd International Conference on Machine Learning, ACM (2006) 65–72
- [10] DeGroot, M.H.: Optimal Statistical Decisions. John Wiley & Sons (1970) Republished in 2004.
- [11] Kääriäinen, M.: Active learning in the non-realizable case. In: Proceedings of the 17th International Conference on Algorithmic Learning Theory. (2006)
- [12] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* **55**(1) (1997) 119–139
- [13] Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Volume 57 of *Monographs on Statistics & Applied Probability*. Chapman & Hall (November 1993)
- [14] Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 441–448

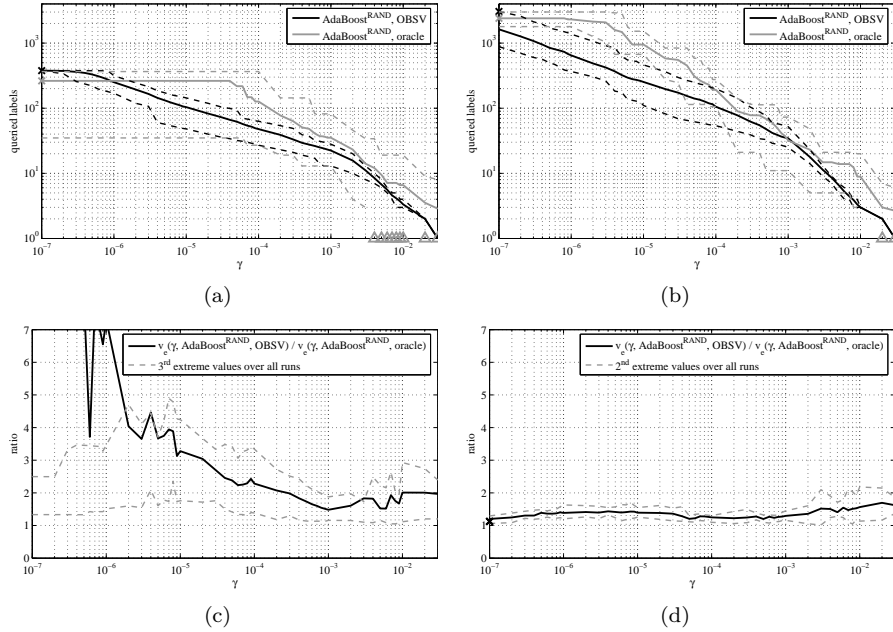


Figure 2: Results for **random sampling** on the **wdbc** (left column) and the **spam** data (right column) as obtained from the 15 (**wdbc**) and 9 (**spam**) runs of AdaBoost with 100 decision stumps. The first row (a), (b), plots the average stopping times from OBSV and the oracle as a function of the labelling cost  $\gamma$ . For each  $\gamma$  the extreme values from all runs are denoted by the dashed lines. The second row, (c), (d), shows the corresponding average ratio in  $v_e$  over all runs between OBSV and the oracle, where for each  $\gamma$  the 3<sup>rd</sup> (**wdbc**) / 2<sup>nd</sup> (**spam**) extreme values from all runs are denoted by the dashed lines. Note a zero value on a logarithmic scale is denoted by a cross or by a triangle. Note for **wdbc** and smaller values of  $\gamma$  the average ratio in  $v_e$  sometimes exceeds the denoted extreme values due to a zero test error occurred in one run.

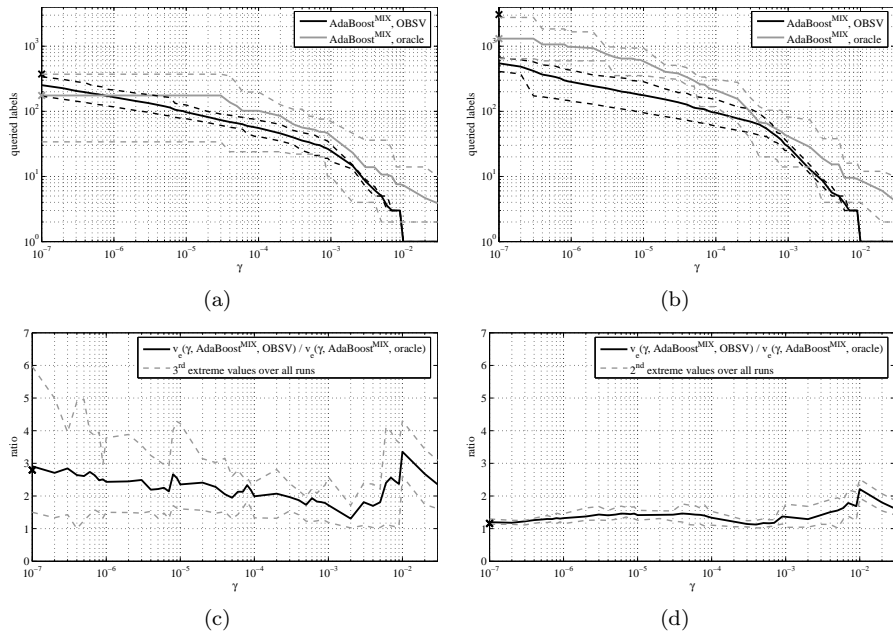


Figure 3: Results for **mixed sampling** on the **wdbc** (left column) and the **spam** data (right column) as obtained from the 15 (**wdbc**) and 9 (**spam**) runs of AdaBoost with 100 decision stumps. The first row (a), (b), plots the average stopping times from OBSV and the oracle as a function of the labelling cost  $\gamma$ . For each  $\gamma$  the extreme values from all runs are denoted by the dashed lines. The second row, (c), (d), shows the corresponding average ratio in  $v_e$  over all runs between OBSV and the oracle, where for each  $\gamma$  the  $3^{\text{rd}}$  (**wdbc**) /  $2^{\text{nd}}$  (**spam**) extreme values from all runs are denoted by the dashed lines. Note a zero value on a logarithmic scale is denoted by a cross.

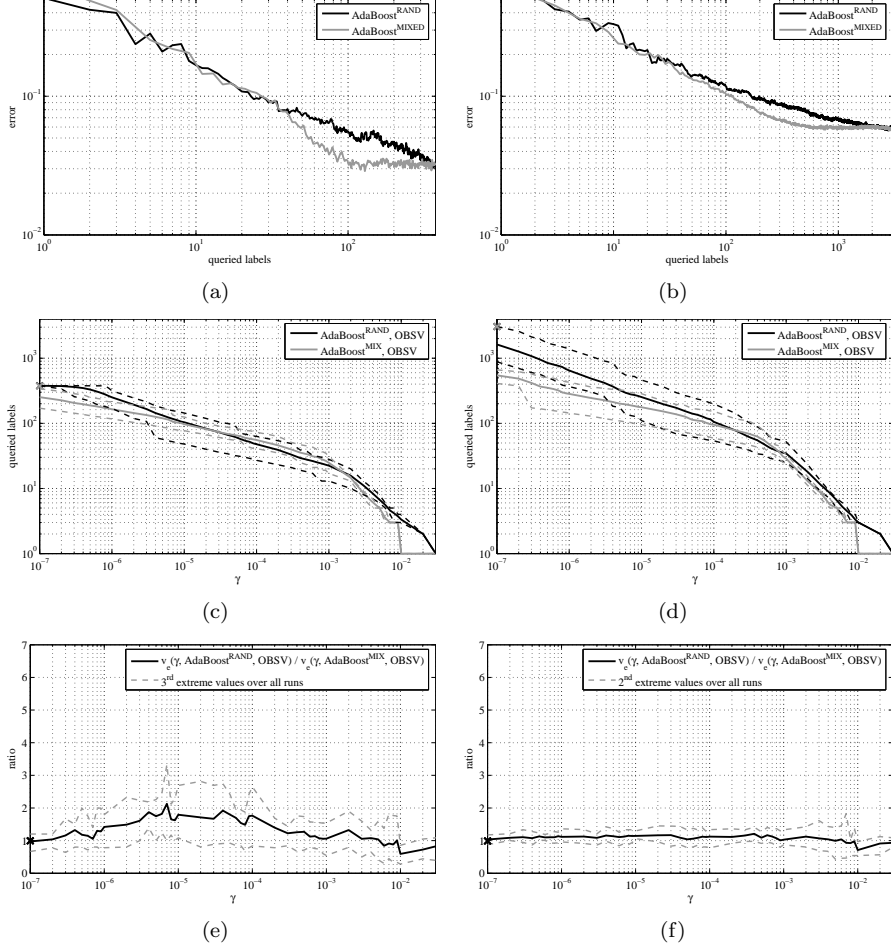


Figure 4: Results comparing random (**RAND**) and mixed (**MIX**) sampling on the **wdbc** (left column) and the **spam** data (right column) as obtained from the 15 (**wdbc**) and 9 (**spam**) runs of AdaBoost with 100 decision stumps. The first row (a), (b), shows the test error of each sampling strategy averaged over all runs. The second row (a), (b), plots the average stopping times from OBSV and the oracle as a function of the labelling cost  $\gamma$ . For each  $\gamma$  the extreme values from all runs are denoted by the dashed lines. The third row, (c), (d), shows the corresponding average ratio in  $v_e$  over all runs between OBSV and the oracle, where for each  $\gamma$  the 3<sup>rd</sup> (**wdbc**) / 2<sup>nd</sup> (**spam**) extreme values from all runs are denoted by the dashed lines. Note a zero value on a logarithmic scale is denoted by a cross.