Noname manuscript No.
(will be inserted by the editor)

Optimised Probabilistic Active Learning (OPAL)

For Fast, Non-Myopic, Cost-Sensitive Active Classification

Georg Krempl · Daniel Kottke · Vincent Lemaire

Received: date / Accepted: date

Abstract In contrast to ever increasing volumes of automatically generated data, human annotation capacities remain limited. Thus, fast active learning approaches that allow the efficient allocation of annotation efforts gain in importance. Furthermore, cost-sensitive applications such as fraud detection pose the additional challenge of differing misclassification costs between classes. Unfortunately, the few existing cost-sensitive active learning approaches rely on time-consuming steps, such as performing self-labelling or tedious evaluations over samples.
We propose a fast, non-myopic, and cost-sensitive probabilistic active learning approach for binary classification. Our approach computes the expected reduction in misclassification loss in a labelling candidate's neighbourhood. We derive and use a closed-form solution for this expectation, which considers the possible values of the true posterior of the positive class at the candidate's position, its possible label realisations, and the given labelling budget.
The resulting myopic algorithm runs in the same linear asymptotic time as uncertainty sampling, while its non-myopic counterpart requires an additional factor of $O(m \cdot \log m)$ in the budget size. The experimental evaluation on several synthetic and real-world data sets shows competitive or better classification performance and runtime, compared to several uncertainty sampling- and error-reduction-based active learning strategies, both in cost-sensitive and cost-insensitive settings.

Keywords Active Learning · Non-Myopic · Cost-Sensitive · Unequal Misclassification Costs · Misclassification Loss · Imbalanced Data · Uncertainty Sampling · Error Reduction

G. Krempl · D. Kottke
KMD Lab, University Magdeburg, Germany
Tel.: +49-391-6758173
Fax: +49-391-6718110
E-mail: [georg.krempl|daniel.kottke]@iti.cs.uni-magdeburg.de

V. Lemaire
Orange Labs, France
Tel.: + 33-296-053107
Fax: + 33-296-051956
E-mail: vincent.lemaire@orange.com

1 Introduction

The volume of automatically generated data increases constantly [13] but human annotation capacities remain limited. Learning from large pools or fast streams of unlabelled data, yet scarce and expensive labelled data, requires the development of fast and efficient active learning algorithms [15, 20]. Such algorithms actively construct a training set, rather than passively processing a given one. Their objective is to select among the unlabelled instances (candidates) the ones for labelling that are deemed to maximise the classification performance the most [32], thus focusing annotation efforts to the most valuable candidates. While *fast* active learning itself poses a challenge, an even bigger one is *cost-sensitivity* [1] [22], where misclassification costs differ between classes, as for example in the diagnosis of rare but dangerous ailments in patients [2], where classifying a sick patient as healthy incurs more severe consequences than classifying a healthy one as sick.

We address these challenges by presenting a novel, fast probabilistic active learning approach, which is suitable for binary classification, both under equal and non-equal misclassification costs. This probabilistic [18, 19] active learning approach computes the expected performance gain, thereby considering both a candidate's *label realisation* and the *true posterior* of the positive class in the candidate's neighbourhood. The latter directly incorporates the likelihoods of different possible posteriors under the already labelled data. This advances most state-of-the-art decision-theoretic active learning literature (e.g. [11, 14]), which considers solely the most likely (or most pessimistic) posterior.

We make three important contributions beyond [18, 19] and other works: First, we optimise label selection for the minimisation of misclassification loss, a *cost-sensitive* performance measure [16]. Second, we derive a *fast, closed-form solution* for calculating the probabilistic gain of an instance. We show that this yields a myopic active learning approach with the same *linear asymptotic computational time* as uncertainty sampling, which is one of the fastest available approaches [32, p. 64]. Third, we propose a *non-myopic* extension of our *optimised probabilistic active learning (OPAL)* approach, where the myopic, isolated view on the value of each label is exchanged for considering the possible remaining number of similar label acquisitions under a given budget. We show that this provides an advantage in particular in cost-sensitive settings, while it requires solely additional time of a factor $O(m \cdot \log m)$ of the budget size. In practical applications, neither this budget size, which for example results from limited human annotation capacities, nor the misclassification costs, which for example are determined by economic consequences as in the German Credit dataset [9], do constitute tunable parameters. Our approach is simple to implement, and neither requires maintaining an evaluation set, nor self-labelling. Experimental evaluation shows its competitiveness in classification performance and speed, compared to other cost-sensitive and cost-insensitive active learning approaches.

The rest of this paper is organised as follows: first, Section 2 provides the necessary background and discusses the related work. Our OPAL-approach is presented in Section 3. The results of its experimental evaluation are reported in Section 4, comparing it to several active learning strategies, including error-reduction and uncertainty sampling approaches specialised for cost-sensitive settings.

[1] Some authors use *cost-sensitive* for differing label acquisition costs between candidates[23].

2 Background and Related Work

Our work addresses cost-sensitive active learning for binary classification. Given the existing overviews on the various existing cost-**in**sensitive approaches in recent surveys such as [32], [12], [6] and [31], we focus on the most related approaches and start with cost-insensitive approaches before moving to cost-sensitive ones.

In expected *error reduction* (ER) [29, 7], the expected error upon incorporating a candidate into the training set is calculated. This is done by simulating for each of its possible label realisations the classifier update, and calculating the resulting error on an evaluation set (e.g. using the set of already labelled instances). In contrast to earlier work [7], the error reduction approach in [29] is usable with any classifier, and directly optimises a user-specified classification performance measure. Nevertheless, ER requires reliable estimates for the true posteriors [3], which are difficult to obtain in early learning stages, where solely few labels are available. Therefore, several regularisation approaches, such as Beta priors, have been explored [3]. This family of approaches is known (see e.g. [32]) to yield good results, but it is slow, requiring an asymptotic runtime of $O(|\mathcal{V}| \cdot |\mathcal{U}|)$, where $\mathcal{V}$ is the evaluation set and $\mathcal{U}$ the pool of candidates.

A fast active learning approach is *uncertainty sampling* (US) [21], which has an asymptotic time complexity of $O(|\mathcal{U}|)$ and is usable on fast data streams [37]. It employs so-called *uncertainty measures* as proxies for a candidate's impact on the classification performance, and the candidate with the highest uncertainty is selected for labelling. In the seminal work of [21], a probabilistic classifier is used on a candidate to compute the posterior of its most likely class. The absolute difference between this posterior estimate and 0.5 is used as uncertainty measure (lower values denoting higher uncertainty). In addition to this confidence-based uncertainty measure, other measures are common as well [32], like entropy or the margin between a candidate and the decision boundary. Similar to the issue of the true posterior above, a known drawback [36] of US is that these proxies do not consider the number of similar instances on which the posterior estimates are made or the decision boundaries are drawn. The reported results of empirical evaluations are somewhat inconclusive, with some authors (e.g. [3] or [30]) reporting for US on some data sets even worse performance than random sampling.

Our recently [19, 18] proposed *probabilistic active learning* (PAL) approach combines the qualities of uncertainty sampling and error reduction, namely being fast and optimising directly a performance measure. Following a smoothness assumption [4], our approach uses probabilistic estimates for summarising the labelled information in a candidate's neighbourhood and evaluating the impact of acquiring a label therein. This impact is expressed by the expected performance gain (the so-called probabilistic gain), measured in terms of an user-defined point classification performance measure [27] like accuracy. Expectation is not only done over the possible realisations of a candidate's label as in error reduction, but also over the true posterior in the candidate's neighbourhood. PAL then selects the candidate that in expectation improves the classification performance the most within its neighbourhood. PAL runs in the same asymptotic time $O(|\mathcal{U}|)$ as uncertainty sampling and showed good results in cost-insensitive classification experiments [19], yet its suitability for cost-sensitive applications is an open question.

Cost-sensitive learning [22] is a particular challenging task for active learning algorithms, where misclassification costs are not equal among different classes.

This occurs for example in fraud detection [9], where positives are rare, but misclassifying them (i.e. producing a false negative) is more costly than misclassifying a negative instance as positive. The objective is then to minimise the misclassification loss [16], i.e. the cost-weighted sum of false positives and false negatives. A related, yet different problem is that of skewed or imbalanced class prior distributions (see [17] for an overview), where one class is far less frequent than the other. This latter problem is addressed in passive classification (where labels are known) by resampling [5,2], i.e. oversampling the minority or undersampling the majority class. In [2], active-learning-based approaches for resampling are reviewed. However, while resampling strategies are useful for creating a balanced training sample, they do not directly address the former problem of cost-sensitive classification itself. Furthermore, the reported empirical results in [9,24] suggest that their suitability for cost-sensitive classification is highly classifier dependent. Thus, we focus on approaches that directly address cost-sensitive classification.

In *passive* classification, unequal misclassification costs are addressed by using classification rules that minimise the conditional risk [8]. A corresponding *active* learning strategy is to use cost-sensitive measures for label selection. This is done in the cost-sensitive variant of ER [25], where misclassification loss is used as error measure, and varying label acquisition costs between instances are considered. Nevertheless, it inherits the slow runtime of ER and its issues associated with the ignorance of the true posterior. For query-by-committee approaches, which use the disagreement between an ensemble of classifiers as a proxy for a candidate's value, [33] proposes a class-weighted, vote entropy-based measure as disagreement metric. However, this approach is specific for natural language processing, where active selection is between given conglomerates of labels.

Uncertainty measures can be made cost-sensitive by weighting posterior estimates with class-specific misclassification costs [22]. However, an active learning component might induce a sampling bias, such that with additional labels the posterior estimates deviate further from the true posterior [22]. This poses a problem especially in cost-sensitive classification tasks, where reliable posterior estimates are required to determine the misclassification-loss optimal decision boundary. [22] addresses this by proposing a so-called cost-sensitive uncertainty sampling approach that performs self-labelling of all remaining unlabelled instances after each label request. This aims at de-biasing the training sample for a cost-sensitive classifier, but also increases the asymptotic time complexity to $O(|\mathcal{U}|^2)$. To the best of our knowledge, no direct empirical comparison between the approaches of [25] and [22] has been published yet. Furthermore, they share another shortcoming in addition to requiring time-consuming steps: they are myopic, as they evaluate the impact of the next label acquisition without considering the remaining labelling budget. Nevertheless, as already stated in early works on active learning [29], the optimal query may very well depend on this remaining budget, which defines how many additional label requests will follow. Thus, extending active learning approaches to become non-myopic (also called far-sighted) is considered relevant [35, 34]. Vijayanarasimhan et al. [34] proposed a far-sighted cost-sensitive active learning method for support vector machines that chooses a set of instances out of the pool of unlabelled candidates incorporating the individual labelling costs into the SVM's objective function. Zhao et al. [35] select a set of instances greedily based on expected entropy reduction. They furthermore suggest to be near-optimal and define a stopping criterion for the active learning process.

3 Optimised Probabilistic Active Learning (OPAL)

We address fast active learning for binary classification in a cost-sensitive environment, where the costs τ of a false positive classification potentially differ from that of a false negative one $(1-\tau)$. The objective of our approach is to select from the pool of unlabelled candidates the one that reduces the misclassification loss [16] the most, once it is labelled and incorporated into the training set.

In the next Subsection 3.1, we provide the detailed modelling and derivation of our probabilistic performance gain estimate (G_{OPAL}), the pseudo-code[2] and a numeric example. This is followed by a discussion of OPAL's properties (Subsection 3.2), in particular under varying misclassification cost ratios and budgets. For convenience, we summarise in Table 3.1 the notation that is subsequently used.

3.1 Modelling and Derivation of G_{OPAL}

Our approach follows a *smoothness assumption* (see ch. 1.2.1, p. 7 in [4]), such that neighbouring positions in the feature space are assumed to have similar posteriors. Given a labelling candidate $(x, .)$ from a pool of unlabelled instances $\mathcal{U}$, and a set $\mathcal{L}$ of already labelled instances (x, y), our approach needs to assess how well its neighbourhood has been explored, i.e. to count the number of already labelled instances that are similar to the candidate, in order to further assess the value of additional labels therein. Estimates on the *posterior probabilities* $\Pr(y|x)$ are *not sufficient*, as their normalisation cancels out the *absolute number* of labels, keeping solely information on the proportion of each class. Therefore, we resort to the unnormalised values. That is, we use the *absolute frequencies* for the number of labels of each class in the candidate's neighbourhood.

We differentiate between two neighbourhood concepts: The first, disjoint one applies to categorical or pre-clustered data. Such data allows to count the number $LC(x, \mathcal{L})$ of labelled instances that are similar to the candidate w.r.t. their features (or assigned cluster). These label counts for x are summarised by its *label statistics* $\text{ls} = (n, \hat{p})$, a tuple consisting of the absolute number n of labels in a candidate's neighbourhood, and the share $\hat{p}$ of positives therein. The second concept of smooth, continuous neighbourhoods corresponds for example to numerical data, where the influence of instances increases with the similarity of their features. In analogy to counts in the first case, we use frequency estimates in this second case. Using probabilistic classifiers that are modified to return unnormalised estimates for the absolute frequencies is one option. We recommend to use generative probabilistic classifiers like Naive Bayes rather than discriminiative ones like logistic regression. The information on the labelled data kept by the former by modelling $\Pr(X, Y)$ and $\Pr(X)$ allows to compute the label statistics directly. Furthermore, generative classifiers converge with fewer labels, as shown in [26], which is important in active learning contexts. If these classifiers are not available, we propose to use Gaussian kernel frequency estimation (here, Σ is the bandwidth matrix):

$$LC(x, \mathcal{L}) \approx KFE(x, \mathcal{L}) = \sum_{x_i \in \mathcal{L}} \exp\left(-\frac{1}{2} \cdot (x - x_i)' \Sigma^{-1} (x - x_i)\right) \qquad (1)$$

[2] Implementations are available on our companion website: `kmd.cs.ovgu.de/res/opal`

Table 1 Used Symbols and Notation

Symbol	Description	Reference
Input Data:		
x	Feature vector of an instance	p. 5
$y \in \{0,1\}$	Class label of an instance (0=neg., 1=pos.)	p. 5
$\mathcal{U} = \{(x,.)\}$	Pool of unlabelled instances	p. 5
$\mathcal{L} = \{(x,y)\}$	Pool of labelled instances	p. 5
Variables Imposed by the Application Domain:		
$\tau \in [0,1]$	Cost of each false positive classification	p. 5, p. 9 Eq. 16
$m \geq 0$	Budget for the candidate's neighbourhood	p. 7
Variables within the Neighbourhood of a Candidate $(x,.)$*:*		
$d_x \geq 0$	Density weight (w.r.t. all instances in $\mathcal{U} \cup \mathcal{L}$)	p. 6 Eq. 2
g_x	Density weighted optimised probabilistic gain	p. 11
$\text{ł}s = (n, \hat{p})$	Label statistics with:	p. 5
n	Total number of already obtained labels	
$\hat{p}$	Share of positives therein (a posterior estimate)	
$k \in \{0, \cdots, m\}$	Number of positives among future label realisations	p. 7
$p \in [0,1]$	True posterior probability of the positive class	p. 7
Functions:		
$L(p\|\text{ł}s)$	Likelihood of a possible true posterior	p. 7 Eq. 7
$\omega_p(S_{\text{label}}) \in [0,1]$	Normalised likelihood of a possible true posterior	p. 8, Eq. 10–12
$\Gamma(z)$	Legendre's gamma function, see p. 206–208 in [28]	p. 7
$\text{I}_{ML}(n, \hat{p}, \tau, m, k)$	Integral, proportional to the expected performance	p. 11 Eq. 33
$G_{\text{OPAL}}(n, \hat{p}, \tau, m)$ $\in [-1,1]$	Optimised probabilistic gain, i.e. a candidate's exp. average misclassification loss reduction	p. 11 Eq. 36

Based on this, we derive the total number of labels $n = LC(x, \mathcal{L})$ and the share of positives therein $\hat{p} = LC(x, \mathcal{L}_+)/LC(x, \mathcal{L})$, where $\mathcal{L}_+$ is the subset of labelled positive instances. The tuple $\text{ł}s = (n, \hat{p})$ constitutes the label statistics of x's neighbourhood. Using Eq. 1, we also derive the density in the candidate's neighbourhood

$$d_x = \frac{LC(x, \mathcal{L} \cup \mathcal{U})}{|\mathcal{L} \cup \mathcal{U}|} \tag{2}$$

This serves as a weight for the importance of the classification performance within this neighbourhood, as compared to other regions in feature space. Therefore, we later weight the average misclassification loss reduction by this density-weight. It is useful to precompute d_x for all candidates in the pool, as $\mathcal{L} \cup \mathcal{U}$ is static in the pool-based active learning scenario.

3.1.1 A Non-Myopic, Cost-Sensitive Probabilistic Gain

Following our recently [18,19] proposed probabilistic active learning approach, we use a candidate's label statistics $\text{ł}s$ to compute its probabilistic gain, which corresponds to the expected gain in classification performance from acquiring the candidate's label. This is done by first modelling both, the candidate's label realisation y and the true posterior p of the positive class in its neighbourhood, as random variables, and computing the expectation over both variables simultaneously, using the normalised likelihood given the label statistics. The resulting *probabilistic gain* is weighted by the feature density d_x at its position, and the candidate with highest density-weighted probabilistic gain is selected.

However, in contrast to [18,19], our *optimised probabilistic active learning* (OPAL) offers three advantages for fast, cost-sensitive applications: first, it quantifies a candidate's probabilistic gain (its label's value) in terms of *misclassification loss reduction*, which is a cost-sensitive measure. Second, it uses a closed-form solution for computing the probabilistic gain, making it faster. Third, it is *non-myopic*, considering the effect of more than one label acquisition at once. Thus, given a budget that allows to acquire m labels at once within the neighbourhood, we compute the expectation over a set $y_1, y_2, \cdots, y_m$ of label realisations, rather than on a single label realisation y. However, the ordering of labels is irrelevant in this additional training set. By counting the number of positive realisations k in the possible sets, $m+1$ different cases ($k = 0, 1, \cdots, m$) are distinguishable, and the number of positive realisations is a binomial-distributed random variable $K \sim \mathrm{Bin}_{m,p}$. Thus, we perform the expectation over its realisation k and over the true posterior p, rather than over y and p as in [19].

The true posterior in this neighbourhood is unknown, but for its possible values, the likelihoods are calculable by using the frequency estimates from the label statistics. For this, we model the unknown true posterior in this neighbourhood by a Beta-distributed random variable P. Its realisation p is itself the parameter of the Bernoulli distribution that controls the label realisation $y \in \{0, 1\}$ of any instance within the neighbourhood. Consequently, for any set of m label realisations in the neighbourhood, the number k of positives therein is the realisation of a Binomial-distributed random variable K:

$$P \sim \mathrm{Beta}_{n\cdot\hat{p}+1,\, n\cdot(1-\hat{p})+1} \tag{3}$$

$$Y \sim \mathrm{Bernoulli}_p = \mathrm{Ber}_p \tag{4}$$

$$K \sim \mathrm{Binomial}_{m,p} = \mathrm{Bin}_{m,p} \tag{5}$$

We will denote the probability (density) functions (pdf's) of the above distributions by $\mathrm{Beta}_{\alpha,\beta}()$, $\mathrm{Ber}_p()$, and $\mathrm{Bin}_{m,p}()$, respectively. For computing the binomial coefficient in $\mathrm{Bin}_{m,p}(k) = \binom{m}{k} \cdot p^k \cdot (1-p)^{m-k}$, as well as in the subsequent equations below, we use the generalised binomial coefficient for non-integer arguments, and the gamma function $\Gamma(z)$ as defined by Legendre[3]:

$$\binom{m}{k} = \frac{\Gamma(m+1)}{\Gamma(k+1)\cdot\Gamma(m-k+1)} \tag{6}$$

The true posterior's Beta distribution above is the result of its normalised likelihood, given the already observed labels, as we will show below. According to Eq. 5, the likelihood of a true posterior p given the data summarised in $\mathcal{L}$ corresponds to the probability mass function $\mathrm{Bin}_{n,p}(n\hat{p})$:

$$L(p|\mathcal{L}) = L(p|(n,\hat{p})) = \mathrm{Bin}_{n,p}(n\hat{p}) \tag{7}$$

$$= \frac{\Gamma(n+1)\cdot p^{n\cdot\hat{p}}\cdot(1-p)^{n\cdot(1-\hat{p})}}{\Gamma(n\cdot\hat{p}+1)\cdot\Gamma(n\cdot(1-\hat{p})+1)} \tag{8}$$

Following a Bayesian approach, we consider a prior $g(p)$ for P, and obtain the normalised likelihood

$$\omega_{\mathcal{L}}(p) = \frac{L(p|\mathcal{L})g(p)}{\int_0^1 L(\psi|\mathcal{L})g(\psi)\mathrm{d}\psi} \tag{9}$$

[3] See for example pages 206–208 in [28].

The choice of a suitable prior $g(p)$ depends on our a priori information about the class prior distribution $\Pr(Y = +)$ in the data. Without any a priori information, we chose a uniform prior for P, i.e. $g(p) \sim U(0, 1)$. As a result, $g(p)$ is a constant function, and the integral in the denominator sums up to $(1+n)^{-1}$, yielding $(1+n)$ as normalising constant:

$$\omega_{\text{ł}\text{s}}(p) = (1 + n) \cdot L(p|\text{ł}\text{s}) \tag{10}$$

Expanding this using Eq. 7, and setting $(1 + n) \cdot \Gamma(n + 1) = \Gamma(n + 2)$, we obtain precisely the probability function of the Beta-distribution:

$$\omega_{\text{ł}\text{s}}(p) = \frac{\Gamma(n + 2) \cdot p^{n \cdot \hat{p}} \cdot (1 - p)^{n \cdot (1-\hat{p})}}{\Gamma(n \cdot \hat{p} + 1) \cdot \Gamma(n \cdot (1 - \hat{p}) + 1)} \tag{11}$$

$$= \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} \cdot p^{\alpha-1} \cdot (1 - p)^{\beta-1} = \text{Beta}_{\alpha,\beta}(p) \tag{12}$$

Here, we rewrite in the last step the arguments of the Γ-functions and obtain $\alpha = n \cdot \hat{p} + 1$ and $\beta = n \cdot (1 - \hat{p}) + 1$. These parameters for the Beta-distribution have a correspondence in the positive and negative labels in the candidate's neighbourhood: with each positive label therein, α increases by one, while with each negative, β increases by one. Thus, the normalised likelihood expressed by the Beta-distribution is a uniform distribution if no labels are available. However, if labels are available, its peak around $\hat{p}$ becomes more and more distinct with an increase in the number of available labels.

In contrast to [19], we do the expectation in OPAL over k and p, rather than over y and p, yielding the candidate's probabilistic gain (G_{OPAL}), defining the expected change of the performance measure for its neighbourhood in average per additional label:

$$G_{\text{OPAL}}(\text{ł}\text{s}, \tau, m) = \frac{1}{m} \cdot \mathtt{E}_p \left[\mathtt{E}_k \left[\text{gain}_p(\text{ł}\text{s}, k, m) \right] \right] \tag{13}$$

$$= \frac{1}{m} \cdot \int_0^1 \text{Beta}_{\alpha,\beta}(p) \cdot \sum_{0 \le k \le m} \text{Bin}_{m,p}(k) \cdot \text{gain}_p(\text{ł}\text{s}, k, m) \, \mathrm{d}p \tag{14}$$

Here, $\text{gain}_p(\text{ł}\text{s}, k, m)$ is the performance gain within the neighbourhood with label statistics $\text{ł}\text{s}$ and true posterior p, given that k among m additional label realisations are positive. Using a point performance measure $\text{perf}_p(\hat{p})$ that calculates the classification performance under a posterior estimate $\hat{p}$ and a true posterior p, the performance gain is written as difference between future and current performance:

$$\text{gain}_p(\text{ł}\text{s}, k, m) = \text{perf}_p\left(\frac{n\hat{p} + k}{n + m}\right) - \text{perf}_p(\hat{p}) \tag{15}$$

We address active learning for cost-sensitive binary classification tasks, where $\tau \in [0; 1]$ indicates the cost for each false positive instance, and $1 - \tau$ is the corresponding cost for each false negative one, assuming zero costs for correct classifications. A point performance measure [27] for this setting is misclassification loss [16], which is the product of the misclassification cost matrix and the confusion

matrix. Within a neighbourhood with true posterior p and a classification rule classifying a share of q instances therein as positive[4], the misclassification loss is

$$MLoss(p,q) = p \cdot (1-q) \cdot cost_{FN} + (1-p) \cdot q \cdot cost_{FP} = \tag{16}$$
$$p \cdot (1-q) \cdot (1-\tau) + (1-p) \cdot q \cdot \tau = q \cdot (\tau - p) + p \cdot (1-\tau) \tag{17}$$

Thus, given p and τ, the misclassification loss is a linear function of $q \in [0;1]$. It has a positive slope for $p < \tau$, and a negative for $p > \tau$. Due to the positive slope, it is optimal to set $q = 0$ in the former case (and $q = 1$ in the latter), in order to minimise the loss function. Thus, the cost-optimal decision is:

$$q^* = \begin{cases} 0 & p < \tau \\ 1-\tau & p = \tau \\ 1 & p > \tau \end{cases} \tag{18}$$

One could argue that in the case $\tau = p$ the choice of q^* is an arbitrary one, as the first factor $q \cdot (\tau - p)$ is zero, meaning equal loss of $p^2 = \tau^2$ for all choices of q^*. However, in order to obtain a *consistent* classification rule, one should specify the assignment $q^* = 1 - \tau$ at ties, rather than simply replacing one strict inequality condition in Eq. 18 with a non-strict one. This is illustrated when studying the classification under extreme values for τ. For example, if $\tau = 0$, false positives do not cost anything, while false negatives are very expensive. A cost-optimal classification rule should thus classify every instance as positive, i.e. q^* should be one for all possible p. While cases of $p \in]0;1]$ are covered by the third clause in Eq. 18, the second clause must return $q^* = 1$ for cases of $p = 0$. Vice versa, if $\tau = 1$ this second clause must return $q^* = 0$. Thus, it should be set to $1 - \tau$ (or $1 - p$, equivalently). As the true posterior p is not directly observable, the counted observed share $\hat{p}$ of positives is used as proxy instead. Thus, ties might occur frequently enough to consider this as relevant.

Given p and τ, using negated misclassification loss (Eq. 16) under cost-optimal classification (Eq. 18), we derive a performance measure suited for Eq. 15:

$$perf_{p,\tau}(\hat{p}) = -ML_{p,\tau}(\hat{p}) = -\begin{cases} p \cdot (1-\tau) & \hat{p} < \tau \\ \tau \cdot (1-\tau) & \hat{p} = \tau \\ \tau \cdot (1-p) & \hat{p} > \tau \end{cases} \tag{19}$$

Intentionally, we do not include the density of the neighbourhood here, to measure the effect of this misclassification loss reduction for the whole data set, because we intend to separate data set specific information from this value. Of course, $ML_{p,\tau}(\hat{p})$ should have been multiplied with the neighbourhood's density d_x, but this factor can be delivered to the very left side of the whole formula.

Plugging this into Eq. 14 yields the probabilistic misclassification loss reduction

$$G_{\mathrm{OPAL}}(\mathit{ls}, \tau, m) = \frac{1}{m} \cdot \int_0^1 \mathrm{Beta}_{\alpha,\beta}(p) \sum_{k=0}^{m} \mathrm{Bin}_{m,p}(k) \left(ML_{p,\tau}(\hat{p}) - ML_{p,\tau}\left(\frac{n\hat{p}+k}{n+m}\right)\right) \mathrm{d}p \tag{20}$$

[4] Classification within a neighbourhood is assumed to be indifferently of the precise location within the neighbourhood, i.e. we assume conditional independence of the posterior given the neighbourhood of an instance.

3.1.2 Derivation of The Closed-Form Solution

For deriving a closed-form solution, we split Eq. 20 it into a term for to the expected current performance E_{cur} and another for the expected future performance E_{fut}:

$$G_{\mathrm{OPAL}}(\not{s}, \tau, m) = \frac{1}{m} \cdot (E_{cur} - E_{fut}) \tag{21}$$

The first term E_{cur}, where $m = k = 0$ and $\mathrm{Bin}_{0,p}(0) = 1$, is simplified to:

$$E_{cur} = \int_0^1 \mathrm{Beta}_{\alpha,\beta}(p) \cdot ML_{p,\tau}(\hat{p})\,\mathrm{d}p \tag{22}$$

Expanding the Beta-distributed probability $\mathrm{Beta}_{\alpha,\beta}(p)$ by Eq. 12 and the misclassification loss by the case-by-case formula in Eq. 18, and integrating out yields:

$$E_{cur} = \int_0^1 \frac{\Gamma(n+2) \cdot p^{n\cdot\hat{p}} \cdot (1-p)^{n\cdot(1-\hat{p})}}{\Gamma(n\cdot\hat{p}+1)\cdot\Gamma(n\cdot(1-\hat{p})+1)} \cdot \begin{cases} p\cdot(1-\tau)\,\mathrm{d}p & \hat{p} < \tau \\ \tau\cdot(1-\tau)\,\mathrm{d}p & \hat{p} = \tau \\ \tau\cdot(1-p)\,\mathrm{d}p & \hat{p} > \tau \end{cases} \tag{23}$$

$$= (n+1)\cdot\binom{n}{n\cdot\hat{p}} \cdot \begin{cases} (1-\tau) \cdot \frac{\Gamma(1+n-n\hat{p})\Gamma(2+n\hat{p})}{\Gamma(3+n)} & \hat{p} < \tau \\ (\tau-\tau^2) \cdot \frac{\Gamma(1+n-n\hat{p})\Gamma(1+n\hat{p})}{\Gamma(2+n)} & \hat{p} = \tau \\ \tau \cdot \frac{\Gamma(2+n-n\hat{p})\Gamma(1+n\hat{p})}{\Gamma(3+n)} & \hat{p} > \tau \end{cases} \tag{24}$$

The second term, E_{fut}, in Eq. 21 is:

$$E_{fut} = \int_0^1 \mathrm{Beta}_{\alpha,\beta}(p) \sum_{k=0}^{m} \mathrm{Bin}_{m,p}(k) \cdot ML_{p,\tau}\left(\frac{n\hat{p}+k}{n+m}\right)\mathrm{d}p \tag{25}$$

$$= \sum_{k=0}^{m} \cdot \int_0^1 \mathrm{Beta}_{\alpha,\beta}(p) \cdot \mathrm{Bin}_{m,p}(k) \cdot ML_{p,\tau}\left(\frac{n\hat{p}+k}{n+m}\right)\mathrm{d}p \tag{26}$$

As above, we expand the terms therein, which now include the Binomial-distributed probability $\mathrm{Bin}_{m,p}(k) = \binom{m}{k} \cdot p^k \cdot (1-p)^{m-k}$:

$$E_{fut} = \sum_{k=0}^{m} \int_0^1 \frac{\Gamma(n+2) \cdot p^{n\cdot\hat{p}} \cdot (1-p)^{n\cdot(1-\hat{p})}}{\Gamma(n\cdot\hat{p}+1)\cdot\Gamma(n\cdot(1-\hat{p})+1)} \cdot \tag{27}$$

$$\cdot \binom{m}{k} \cdot p^k \cdot (1-p)^{m-k} \cdot ML_{p,\tau}\left(\frac{n\hat{p}+k}{n+m}\right)\mathrm{d}p \tag{28}$$

$$= (n+1)\cdot\binom{n}{n\cdot\hat{p}} \cdot \sum_{k=0}^{m} \cdot \mathrm{I}_{ML}(n,\hat{p},\tau,m,k) \tag{29}$$

where I_{ML} is a function of n, $\hat{p}$, τ, m, and k, containing the integral and proportional to the expected performance, which is integrated out as follows:

$$\mathrm{I}_{ML}(n,\hat{p},\tau,m,k) = \tag{30}$$

$$= \int_0^1 \binom{m}{k} \cdot p^{n\cdot\hat{p}+k} \cdot (1-p)^{n+m-n\cdot\hat{p}-k} \cdot ML_{p,\tau}\left(\frac{n\hat{p}+k}{n+m}\right) \mathrm{d}p \tag{31}$$

$$= \binom{m}{k} \cdot \int_0^1 p^{n\cdot\hat{p}+k} \cdot (1-p)^{n+m-n\cdot\hat{p}-k} \cdot \begin{cases} p\cdot(1-\tau)dp & \frac{n\hat{p}+k}{n+m} < \tau \\ (\tau-\tau^2)dp & \frac{n\hat{p}+k}{n+m} = \tau \\ \tau\cdot(1-p)dp & \frac{n\hat{p}+k}{n+m} > \tau \end{cases} \tag{32}$$

$$= \binom{m}{k} \cdot \begin{cases} (1-\tau) \cdot \frac{\Gamma(1-k+m+n-n\hat{p})\Gamma(2+k+n\hat{p})}{\Gamma(3+m+n)} & \frac{n\hat{p}+k}{n+m} < \tau \\ (\tau-\tau^2) \cdot \frac{\Gamma(1-k+m+n-n\hat{p})\Gamma(1+k+n\hat{p})}{\Gamma(2+m+n)} & \frac{n\hat{p}+k}{n+m} = \tau \\ \tau \cdot \frac{\Gamma(2-k+m+n-n\hat{p})\Gamma(1+k+n\hat{p})}{\Gamma(3+m+n)} & \frac{n\hat{p}+k}{n+m} > \tau \end{cases} \tag{33}$$

Using this in equations 24 and 29, we obtain

$$E_{cur} = (n+1) \cdot \binom{n}{n\cdot\hat{p}} \cdot \mathrm{I}_{ML}(n,\hat{p},\tau,0,0) \tag{34}$$

$$E_{fut} = (n+1) \cdot \binom{n}{n\cdot\hat{p}} \cdot \sum_{k=0}^{m} \cdot \mathrm{I}_{ML}(n,\hat{p},\tau,m,k) \tag{35}$$

and equation 36 to compute the G_{OPAL} in the candidate's neighbourhood:

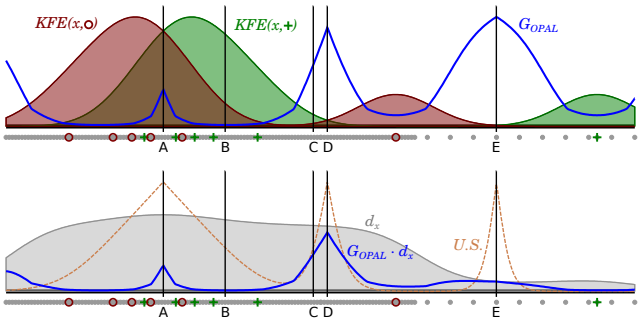
$$G_{\mathrm{OPAL}}(n,\hat{p},\tau,m) = \frac{(n+1)}{m} \cdot \binom{n}{n\cdot\hat{p}} \cdot \left(\mathrm{I}_{ML}(n,\hat{p},\tau,0,0) - \sum_{k=0}^{m} \mathrm{I}_{ML}(n,\hat{p},\tau,m,k)\right) \tag{36}$$

3.1.3 Pseudocode and Numeric Examples

The pseudocode for OPAL in pool-based active learning is given in Figure 1. Lines 2 – 7 iterate over each labelling candidate $(x,.)$ in the pool $\mathcal{U}$. First (line 3), a candidate's label statistics $\mathit{ls} = (n_x, p_x)$ are computed according to Eq. 1. Second (line 4), the density weight d_x is estimated, i.e. the proportion of all labelled and unlabelled instances within the candidate's neighbourhood divided by those in all neighbourhoods, see Eq. 2. In line 5, the optimal m_x^* that maximises the G_{OPAL} (see Eq. 36) is found, using a *logarithmic search* over $m' = 1, 2, \cdots, m$. Density-weighting this maximal G_{OPAL} (line 6) yields g_x. , and the candidate maximising the density-weighted probabilistic gain is returned (line 8).

```
1: function PoolBasedOPAL(𝒰,ℒ,τ,m)
2:     for x ∈ 𝒰 do
3:         (n_x, p̂_x) ← labelstatistics(x, ℒ)
4:         d_x ← densityweight(x, ℒ ∪ 𝒰)
5:         m*_x ← arg max_{m'∈1,2,⋯,m} G_OPAL((n_x, p̂_x), τ, m')
6:         g_x ← G_OPAL((n_x, p̂_x), τ, m*_x) · d_x
7:     end for
8:     return arg max_{x∈𝒰}(g_x)
9: end function
```

Fig. 1 The OPAL Algorithm



	$\hat{p}$	n	d_x
A	.5	6.0	.16
B	.8	3.6	.15
C	.8	0.5	.14
D	.5	0.4	.13
E	.5	0.1	.02

	G_{OPAL}	
	$\cdot 1$	$\cdot d_x$
A	.0278	.00435
B	.0011	.00016
C	.0459	.00617
D	.0737	.00982
E	.0817	.00164

Fig. 2 Visualisation of G_{OPAL}-values for $\tau = 0.5$ on a one-dimensional dataset with labelled (red resp. green dots) and unlabelled (grey dots) data points. The upper plot shows the kernel frequency estimates (KFE) for each class and the corresponding G_{OPAL}-value (blue curve). The lower plot shows the density (grey area) and the density-weighted G_{OPAL}-values (blue curve). Additionally, the negative confidence values from the Uncertainty Sampling approach are plotted for comparison. For exemplary data points (A-E) the corresponding label statistics and the unweighted ($\cdot 1$) and density weighted ($\cdot d_x$) G_{OPAL}-values are given in the tables.

The G_{OPAL}, visualised in Fig. 2, corresponds to the expected *average* reduction in misclassification loss in each subsequent classification[5] in the candidate's neighbourhood. For equal misclassification costs, the G_{OPAL} is proportional to the expected average gain in accuracy and is highest for candidates close to the decision boundary (where $\hat{p} \approx \tau$), like the points D and E (compared to B and C) in Fig. 2. For a very small number n of already obtained similar labels, G_{OPAL} approximates random sampling as $n \to 0$, corresponding to the barely available information. Nevertheless, as n increases (compared to the remaining budget m), the equations above are dominated by the observed posterior $\hat{p}$. Thus, the difference between expected future $\mathrm{I}_{ML}(n, \hat{p}, \tau, m, k)$ and current performance $\mathrm{I}_{ML}(n, \hat{p}, \tau, 0, 0)$ converges towards zero, making candidates in well-explored regions (e.g. A) less valuable than those in unexplored ones (e.g. D,E). In the lower subplot in Fig.2, the points (D, E) show the importance of the density-weighting: Point E is in a less explored but also sparser area than D, thus E has a higher G_{OPAL} (0.0817 vs. 0.0737) but a 6.5-times lower density weight, as improving the performance in its region will effect 6.5 times fewer future classifications. Thus, the density-weighted probabilistic gain of E (0.00164) is lower than that of D (0.00982). In contrast, US neither incorporates the amount of available information (e.g. A vs. D), nor the importance of neighbourhoods (e.g. D vs. E). Note that for unequal misclassification costs, the G_{OPAL} is not symmetric around τ, but rather favours sampling instances from the regions where potentially a more costly error is made. That is, if false positive costs are relatively low compared to false negative ones (e.g. $\tau = 0.1$), misclassification of positives (as false negatives) is expensive compared to the misclassification of negatives. Accordingly, the probabilistic gain is higher in regions where currently instances are classified as negative, as the possible error therein is more expensive. Therefore, our cost-sensitive approach will favour sampling in these regions. A further discussion of the properties of G_{OPAL} is provided in Section 3.2.2, where Figure 3 on page 14 illustrates the shape of this function.

[5] Assuming cost-optimal classification [8], see Eq. 18 in Subsection 3.1.

3.2 Properties of G_{OPAL}

We now briefly discuss the asymptotic (with respect to data set size) computational time complexity of OPAL in Subsection 3.2.1, comparing it to related algorithms for active learning of binary, incremental classifiers, before illustrating the effect of the myopic extension on the probabilistic gain in Subsection 3.2.2.

3.2.1 Computational Complexity

For the non-myopic selection of a candidate from a pool $\mathcal{U}$ of labelling candidates, OPAL iterates first over all candidates in the pool (lines 2 – 7). Each iteration consists of 1) querying label statistics, 2) querying density weights, 3) determining the locally optimal budget, and 4) computing the density-weighted probabilistic gain. The first step requires absolute frequency estimates of labels in the candidate's neighbourhood, similar to the relative frequency estimates needed by entropy or confidence uncertainty measures. These are obtained in constant time by probabilistic classifiers. The second step requires density estimates over all instances, that is over labelled $\mathcal{L}$ and unlabelled $\mathcal{U}$ ones. Precomputing these density estimates once for all later calls of OPAL leads to constant query time, as in the pool-based setting the union $\mathcal{L} \cup \mathcal{U}$ is constant. The third step requires a logarithmic search over all possible $m' \in 1, 2, \cdots, m$, where for each m' the G_{OPAL} is calculated. The latter is done in $O(m)$ time, due to the closed-form solution obtained for Eq. 36. Thus, the third step requires $O(m \log(m))$ time. The fourth step computes the density-weighted probabilistic gain g_x, requiring constant time. For the subsequent selection of the best candidate in line 8, the maximum of g_x as well as the index of the corresponding candidate are kept. Thus, the iteration over the pool is determining the overall asymptotic time complexity of $O(|\mathcal{U}| \cdot m \log(m))$ for the *non-myopic* OPAL, where m is the remaining labelling budget that is in general much smaller than $|\mathcal{U}|$. OPAL's *myopic* counterpart requires asymptotically linear time $O(|\mathcal{U}|)$, as $m = 1$.

In comparison, uncertainty sampling also requires asymptotically linear time $O(|\mathcal{U}|)$, whereas error reduction as discussed in [32] requires $O(|\mathcal{U}| \cdot |\mathcal{V}|)$ time, where $|\mathcal{V}| \approx |\mathcal{U}|$, as $\mathcal{V}$ needs to be a representative sample of the data.

3.2.2 Effect of the Non-Myopic Extension

Besides of being faster and cost-sensitive, OPAL extends PAL by adding the ability of acting non-myopic. The first two properties are the result from using a closed-form solution and misclassification loss as performance measure, and have already been discussed. Thus, we will now focus on the third one, which allows OPAL to consider a budget m when computing the probabilistic gain of a candidate.

We illustrate the usefulness and effect of this extension on the probabilistic gain function in Figure 3. In this figure, the first two columns of plots show the probabilistic gain function (in terms of average misclassification loss reduction) for different label statistics, i.e. combinations of different numbers of already obtained labels n and observed posteriors $\hat{p} = \hat{Pr}(+|x)$. The first column shows the myopic probabilistic gain ($m = 1$), the second the non-myopic one ($m = m^*$), where $m \in \{1, 2, \cdots, 21\}$ is chosen such that the probabilistic gain is maximal. The third column corresponds to the difference (in a logarithmic scale) between the two

probabilistic gains. The three rows correspond to the different misclassification costs $\tau = 0.1$, 0.25, and 0.5. The plots for $\tau = 0.75$ (and $\tau = 0.9$) are not shown, as they are reflection symmetric to those of $\tau = 0.25$ (and $\tau = 0.1$, respectively).

Given equal misclassification costs ($\tau = 0.5$, third row) and a candidate in a neighbourhood, where already two labels (all positive) have been acquired ($n = 2$, $\hat{p} = 1$). In this neighbourhood, a single additional label can not alter the classification decision, thus the probabilistic gain in a myopic setting is zero. This is seen in the left-bottom plot, where the probabilistic gain is zero for $n = 2$, $\hat{p} = 1$ (the corner in the uttermost back). However, if more than one label can be acquired in this neighbourhood, these labels might change the classification. Thus, under a non-myopic setting and equal misclassification costs, the probabilistic gain should be positive. Indeed, the probabilistic gain is $G_{\mathrm{OPAL}} = 0.0274$ for the optimal acquisition of three labels ($m^* = 3$). Correspondingly, the flanks in the centred plots are flatter than in the left ones.

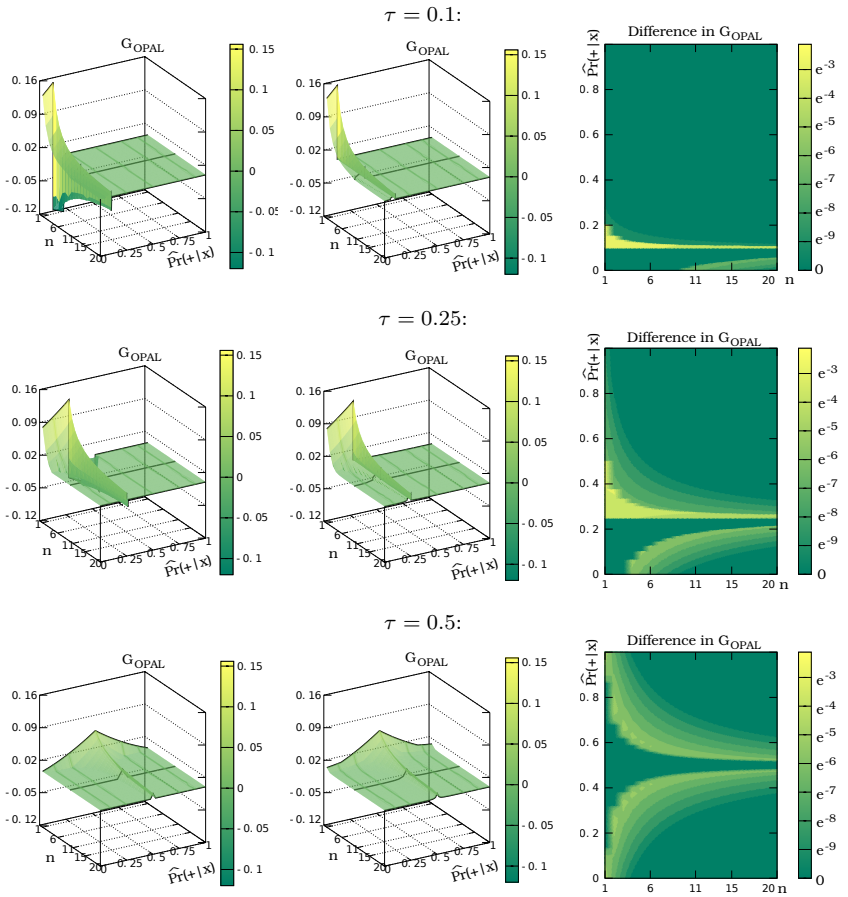


Fig. 3 Plots of the G_{OPAL} as function of observed posterior $\hat{p} = \hat{P}r(+|x)$ and number of labels n for different cost-rations $\tau = 0.1, 0.25, 0.5$ (rows). The left column shows the myopic G_{OPAL}, the centre column shows the non-myopic G_{OPAL}, and the right column shows the difference between the two (in logarithmic scale).

For unequal misclassification costs (e.g. in first row with $\tau = 0.1$, meaning false positives are cheap compared to false negatives), the expected gain from a single additional label might even be negative. This corresponds to situations, where just sufficiently positive labels were acquired within a neighbourhood to classify instances therein as positive (i.e. $\hat{p} = \hat{Pr}(+|x) > \tau$). In the myopic setting, the realisation of the single additional label is binary. If it is positive, it does not alter the classification. If it is negative, it inverts the classification. The latter results in a wrong classification if the true posterior p is actually greater τ, which is very likely, given that the share of positives $\hat{p}$ among the already seen labels was greater τ. Therefore, in the left-upper plot, the myopic probabilistic gain is negative for $n = 1$ and $\hat{p} \in [0.1, 0.2]$, with a negative peak at $G_{\text{OPAL}} = -0.12$ for $\hat{p} = 0.199$.

In contrast to the myopic setting with its binary label realisation, the non-myopic setting uses a rational number: the share of positives among the realisation of additional labels. Thus, the effect of this special case decreases with increasing m, as shown in the upper-centre plot. Nevertheless, for extreme misclassification cost inequalities (e.g. $\tau = 0.1$ or $\tau = 0.9$), the probabilistic gain remains non-positive for all neighbourhoods with $\hat{p} > \tau$, which are therefore not selected for label requests. In contrast, for moderate misclassification cost inequalities (e.g. $\tau = 0.25$ or $\tau = 0.75$), it becomes positive for some neighbourhoods therein, namely those having an observed posterior $\hat{p}$ close to τ. As a consequence, the effect of the non-myopic extension might be more important for situations with moderate misclassification cost inequalities, than for those with either equal misclassification costs or extreme unequal ones. However, this requires empirical evaluation, which we provide in Section 4.3.

Concerning the probabilistic gain function's mode, our numerical experiments indicate it to be an unimodal function of m for a given combination of n, $\hat{p}$ and τ.

4 Experimental Evaluation

We expect our new *cost-sensitive* method OPAL to perform *at least equally well* in terms of resulting classification performance as other (cost-sensitive) active learning approaches, while being *faster* than other cost-sensitive approaches. Furthermore, we expect OPAL to be better than PAL towards the end of the learning process, through its *non-myopic* extension. Therefore, we designed a framework that ensures a fair evaluation of our contributions.

In the first subsection, we describe our evaluation setting, the active learning approaches used in the comparison, the data sets and our framework. In the second subsection, we present and discuss the results of the experimental evaluation. There, we first assess the usefulness of our cost-sensitive extension. Then, we show that OPAL is in most cases superior, both to its myopic counterpart PAL and to other active learning approaches, while having the same asymptotic time complexity as uncertainty sampling.

4.1 Evaluation Settings

4.1.1 Active Learning Algorithms

For experimental evaluation, we use the fast version of *OPAL* described in Sec. 3, which applies a logarithmic search for determining the optimal budget. In pretests, there was no significant difference in classification performance between this approach and another variant of OPAL doing exhaustive search. In addition, we use a cost-sensitive, myopic *PAL* [19] with the presented speed optimisation (here denoted as csPAL). This is the equivalent to OPAL with a fixed budget of $m = 1$.

Furthermore, we use a cost-sensitive variant of *Uncertainty Sampling* [22] (denoted as U.S.) and *Certainty Sampling* [10] (denoted as C.S.), which both optimise confidence[6] (posterior difference to 0.5). Here, the posterior probabilities are calculated from a cost-weighted frequency estimation. Liu et al. [22] proposed to use a self-training approach for uncertainty sampling, such that posterior estimates are optimised for confidence calculation. We denote this extension as U.S. st.

For error reduction, we use the cost-sensitive algorithm proposed by [25] (denoted as Marg) and the non-cost-sensitive version by [3] (denoted as Chap). As a non-myopic representative, we use the method by [35] (denoted as Zhao). As the latter originally needs initial labels, we use a beta-correction for the classifier predictions of 0.001 (like for Chap). This simulates that in each evaluation neighbourhood an equal number of positives and negatives has been seen. In our framework, we always use 40 labels to be acquired by the active learners. Therefore, we disabled the automatic stopping criterion (otherwise learning often stopped far too early). Furthermore, we use random selection (denoted as Rand) as a baseline.

4.1.2 Data Sets

In our experiments, we used 3 synthetic and 5 real data sets (from [1]). Each attribute was scaled to a $[0; 1]$-range, because we use Gaussian Kernel Frequency estimates with a fixed and pre-tuned bandwidth *sigma* (see Section 6.1). The main characteristics (number of instances, number of attributes), such as training and test set size and the bandwidth σ of the Gaussian Kernel, are given in Tab. 2.

[6] We tested confidence- and entropy-based uncertainty measures in pretests and used the one with the best performance over all data sets for the final evaluation.

		Attributes					
Data set	Instances	real	cat.	$\Pr(+)$	|Train|	|Test|	σ
`See`	210	7	-	33 %	160	50	0.2
`Che`	308	2	-	44 %	200	108	0.08
`Che2`	392	2	-	49 %	250	142	0.08
`Ver`	310	6	-	32 %	260	50	0.08
`Mam`	830	2	2	51 %	630	200	0.7
`Sim`	1200	2	-	50 %	800	400	0.08
`YeaU`	1484	8	-	90 %	1000	484	0.1
`Aba`	4177	8	-	50 %	3500	677	0.25

Table 2 Data set characteristics and parameters (number of instances, number of attributes (real-valued, categorical), proportion of positive instances, training set size, test set size, bandwidth for Parzen window classifier) in ascending training set size order.

Two of the synthetic data sets are based on the generator used in [3]. They consist of $4x4$ clusters, arranged in a checker-board formation (on a 2 dimensional feature space). While the clusters are low-density-separated in `Che` (as in [3]), they are adjoined in `Che2`. The third synthetic data set (`Sim`) consists of two normal distributed, overlapping clusters in a two dimensional space. We used this very simple example as a proof of concept for active learning methods.

The real-world data sets are Seeds (`See`), Vertebral (`Ver`), Mammographic mass (`Mam`), Yeast (`YeaU`) and Abalone (`Aba`), see [1]. As show in Table 2, balanced as well as unbalanced class distributions occur. Categorical features (as in `Mam`) have been dichotomised into multiple binary features. In `Mam`, instances with missing values have been removed. For the multi-class dataset `See`, we classified Kama and Canadian vs. Rose; for `Ver`, we used normal vs. abnormal; for `Aba`, we used trees with rings < 10 vs. rings ≥ 10; for `YeaU`, we used MIT vs. the rest. For better comparison to [3] and [19], we used a Parzen window classifier, which is a generative probabilistic classifier as discussed in Section 3.1. However, for some cost-ratios this classifier is not able to discriminate in the data, thus it is classifying all instances into the more expensive class. This is detectable in the bandwidth tuning curves (see Fig. 6.1), when the best σ-value (the one with lowest misclassification loss) is the maximal, uttermost left one. We reported the results on those data-set/cost-ratio-combinations for completeness, but emphasise that the classification performance on such ill-posed learning problems is not meaningful.

4.1.3 Framework

Our framework decouples classification and active learning. All runs behave exactly the same except for the active sampling component, which decides the instance whose label should be acquired next. Thus, we use exactly the same frequency estimates (see Eq. 1) and the same classification algorithm (a Parzen window classifier [3]) with the identical parameters for any active learning strategy during the calculation. Furthermore, we decoupled the classification and evaluation process to ensure, that every active learning method just differs in the set of labelled instances. To get more significant results, we used a cross-validation with random sub-samplings in 100 runs. The training and test set sizes are listed in Table 2.

Here, active learning starts *without* initial labels on the unlabelled training sample, and finishes after 40 label acquisitions (steps). We implemented the framework in Octave/MATLAB, which was parallelised to run on a cluster. Every run uses the same pre-tuned, data set-specific bandwidth, and each of the 40 steps is evaluated on the same, dedicated (labelled) test sample with the same cost-sensitive Parzen window classifier, recording misclassification loss and speed.

The presented learning curves show the arithmetic mean of the misclassification loss over all 100 runs for a given combination of data set, algorithm and cost-ratio.

4.2 Relevance of the Cost-Sensitivity

From OPAL's theoretical characteristics, we expect OPAL to choose the best instances, with respect to a given cost-ratio τ. To evaluate the relevance of this cost-sensitivity empirically, we run OPAL for its G_{OPAL} calculation with 5 different τ values ($\tau \in \{.1, .25, .5, .75, .9\}$), and evaluate its misclassification loss regarding

the true cost-ratio τ^*. If the cost-sensitivity is meaningful and relevant, the curve with $\tau = \tau^*$ should have the best performance compared to all other τ-values.

Figure 4 shows a selection of data sets (columns) and the evaluation cost-ratios τ^* (rows). Each plot shows the learning curves with the classification performance in terms of misclassification loss on its y-axis and the steps of its learning process in the number of already requested labels on the x-axis. The 5 different variants of OPAL with the varying τ are printed in different colours while the correct one is plotted in bold. The results of the other data sets are given in the appendix (see Section 6.2, Fig. 8).

The first interesting fact is that the correct usage of $\tau = \tau^*$ leads to a converging curve, while some wrong ones lead to diverging curves. Thus, ignoring the application-specific cost-ratio results in a low classification performance on these data sets. Furthermore, the curves for neighbouring τ-values behave similarly, especially the curves for $\tau = 0.25$ and $\tau = 0.1$, respectively $\tau = 0.75$ and $\tau = 0.9$.

Comparing the level and velocity of the misclassification curves, the bold ones are mostly superior. The single exception occurs on `Che`, where OPAL selects optimal labels for $\tau = 0.5$ (that is one instance of each cluster), so it performs very well for the other evaluation cost-ratios too. This is due to the very special characteristics (well-separated clusters) of this data set. When the separation between clusters is reduced, as in `Che2`, the effect vanishes. The other exceptions, like in `YeaU` for $\tau* = .1$, occur all on ill-posed learning problems (see Fig. 6.1), where the results are not meaningful.

Summarising the results, the use of the cost-sensitive extension is beneficial, although there are some exceptional cases, where $\tau \neq \tau^*$ achieves better performance due to special structure in the data. It is noteworthy that in a real-world application, τ is not a tunable parameter but rather imposed by the application domain. The results show that ignoring this application-specific cost-ratio will in most cases result in a non-optimal classification performance.

4.3 Comparison Between OPAL and other Active Learning Strategies

This subsection assess whether (1) OPAL's non-myopic extension is beneficial for a given labelling budget, (2) OPAL's performance is superior or at least equal to other active learning approaches, and (3) its time complexity increases solely linearly with training set size (like uncertainty sampling).

For comparison, we use learning curves measuring the performance in terms of misclassification loss as before. The plots for all data sets and algorithms are given in Figures 5 and 6. The best active learning method is the one, that has a fast-converging, low final misclassification loss level. Furthermore, we give a numerical value (see Table 3) for comparing two algorithms directly over all 8 data sets. It is computed as the portion of OPAL being better than the compared algorithm on all 100 runs of all data sets (thus on 800 pairs) for a given cost-ratio (τ^*) and labelling step. We also report the results of **one-sided Wilcoxon signed-rank tests with significance level** 0.001 on these pairs, by indicating a significantly better performance of OPAL by *, and a significantly worse performance by †.

(1) OPAL's non-myopic extension is beneficial
Here, we compare the non-myopic OPAL and the myopic csPAL. Both algorithms

10 labels	OPAL vs.							
acquired	csPAL	U.S.	U.S. st	C.S.	Marg[1]	Chap[1]	Zhao[1]	Rand
$\tau^* = 0.10$	38%†	51%	52%*	62%*	58%*	47%	64%*	54%*
$\tau^* = 0.25$	60%*	66%*	68%*	82%*	76%*	61%*	73%*	66%*
$\tau^* = 0.50$	1%	70%*	74%*	89%*	80%*	62%*	69%*	72%*
$\tau^* = 0.75$	46%	62%*	65%*	81%*	78%*	58%*	60%*	67%*
$\tau^* = 0.90$	41%†	63%*	64%*	70%*	71%*	58%*	60%*	62%*

20 labels	OPAL vs.							
acquired	csPAL	U.S.	U.S. st	C.S.	Marg[1]	Chap[1]	Zhao[1]	Rand
$\tau^* = 0.10$	47%	62%*	70%*	72%*	66%*	56%*	72%*	62%*
$\tau^* = 0.25$	51%*	63%*	75%*	88%*	81%*	62%*	70%*	65%*
$\tau^* = 0.50$	1%	64%*	72%*	92%*	87%*	63%*	69%*	68%*
$\tau^* = 0.75$	53%*	60%*	67%*	86%*	80%*	50%*	48%*	58%*
$\tau^* = 0.90$	42%	61%*	66%*	77%*	75%*	53%*	57%*	62%*

40 labels	OPAL vs.							
acquired	csPAL	U.S.	U.S. st	C.S.	Marg[1]	Chap[1]	Zhao[1]	Rand
$\tau^* = 0.10$	43%	55%*	71%*	75%*	69%*	62%*	69%*	57%*
$\tau^* = 0.25$	56%*	59%*	73%*	89%*	79%*	65%*	69%*	58%*
$\tau^* = 0.50$	4%	61%*	72%*	93%*	89%*	74%*	76%*	62%*
$\tau^* = 0.75$	57%*	64%*	71%*	90%*	81%*	59%*	56%*	54%*
$\tau^* = 0.90$	46%	55%*	63%*	82%*	77%*	57%*	64%*	56%*

Table 3 Percentages of runs over all data sets, where OPAL performs better than its competitor. Significantly better performance is denoted by *, significantly worse performance by †. The used significance level in the **one-sided Wilcoxon signed-rank test** was for both 0.001. Algorithms are marked with [1] if not every data set could be used in the evaluation due to their long execution time.

Data	OPAL	csPAL	U.S.	U.S. st	C.S.	Marg	Chap	Zhao	Rand
See	1.867	0.254	0.206	0.468	0.162	43.535	51.87	254.8	0.015
Che	1.905	0.249	0.201	0.452	0.183	54.897	56.60	319.9	0.016
Che2	1.968	0.261	0.199	0.510	0.198	66.282	69.68	440.7	0.015
Ver	1.987	0.269	0.202	0.653	0.207	71.126	78.66	451.7	0.015
Mam	2.580	0.353	0.268	3.913	0.277	192.86	280.1	1577	0.016
Sim	2.827	0.335	0.239	2.422	0.202	242.98	302.6	1641	0.016
YeaU	2.993	0.379	0.272	9.318	0.260	285.51	499.9	3050	0.017
Aba	7.000	1.001	0.703	136.1	0.706	NaN	NaN	NaN	0.023

Table 4 Average execution time (in seconds), rows ordered in ascending data set size. All differences w.r.t. OPAL are **significant** (level 0.001, one-sided Wilcoxon signed-rank test).

just differ in their available budget size. While OPAL chooses the best G_{OPAL} for a given m-vector, csPAL just considers the very next possible label ($m = 1$).

As already discussed in Section 3.2.2, the learning curves for $\tau^* = 0.5$ of both algorithms are quite similar. Furthermore, the tables state that OPAL is at most in 4% better than csPAL. This value might confuse, but one must be reminded, that this is just the portion of OPAL being *better*. Because there is no significance of being worse, we can derive that OPAL and csPAL behave quite similar (have the same values). However, if the number of already obtained labels is very high, the myopic G_{OPAL} used in csPAL will get zero, resulting in a random-sampling-like behaviour. In contrast, if m is sufficiently large, meaning that still several more

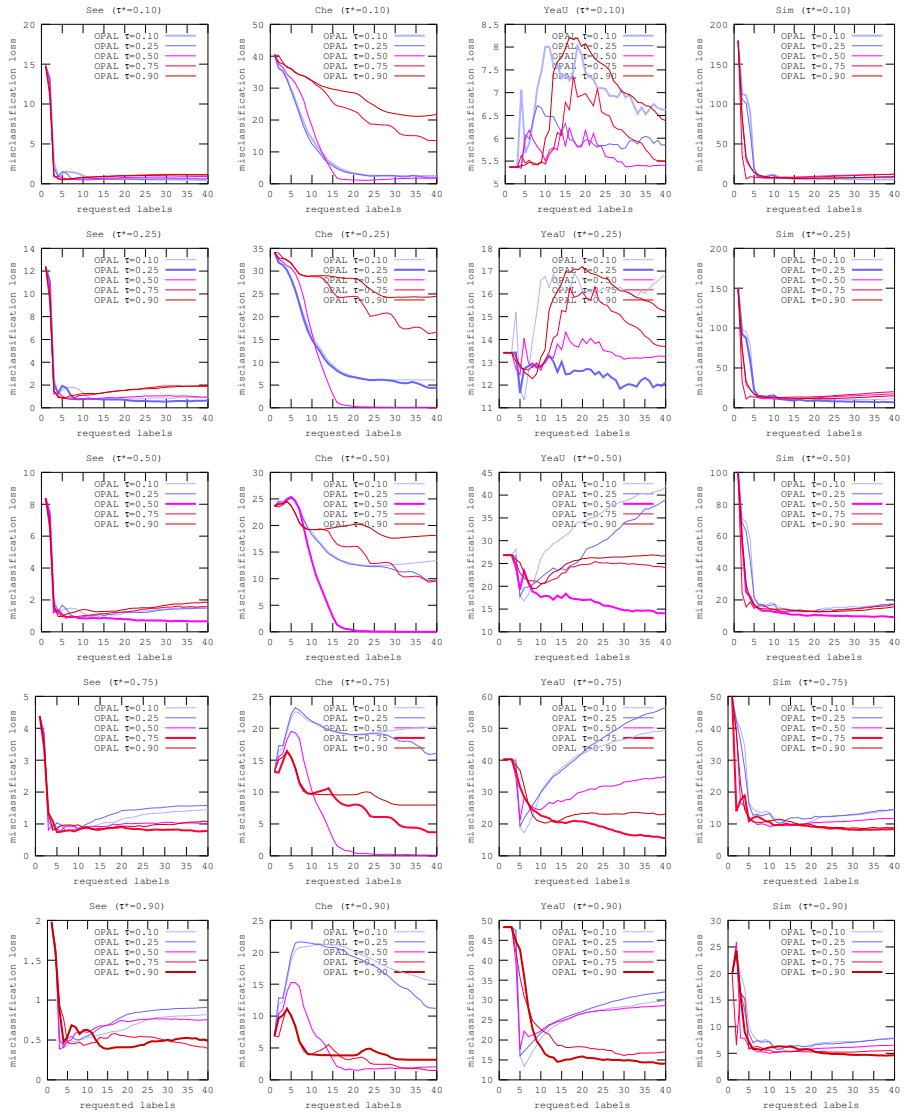


Fig. 4 Misclassification loss curves for OPAL on a selection of data sets with different cost-ratios (τ) for active learning and true cost-ratios (τ^*) for evaluation; curves for correct cost-ratios ($\tau = \tau^*$) are plotted in bold and should be superior; early convergence to very low values is best.

labels will be acquired, the non-myopic variant in OPAL is advantageous, as it will still perform a differentiated selection.

For $\tau^* \neq 0.5$, the non-myopic variant is slightly advantageous in terms of final misclassification loss (e.g. `Mam` for $\tau^* = 0.75$, `YeaU` for $\tau^* = 0.9$ or `See`) or at least performs equally well. Interestingly, while for extreme cost-ratios ($\tau = 0.1$ or $\tau = 0.9$) the non-myopic variant is still advantageous over its myopic counterpart (in 44% or 48% of the cases, see Table 3), the difference is not as big as it is

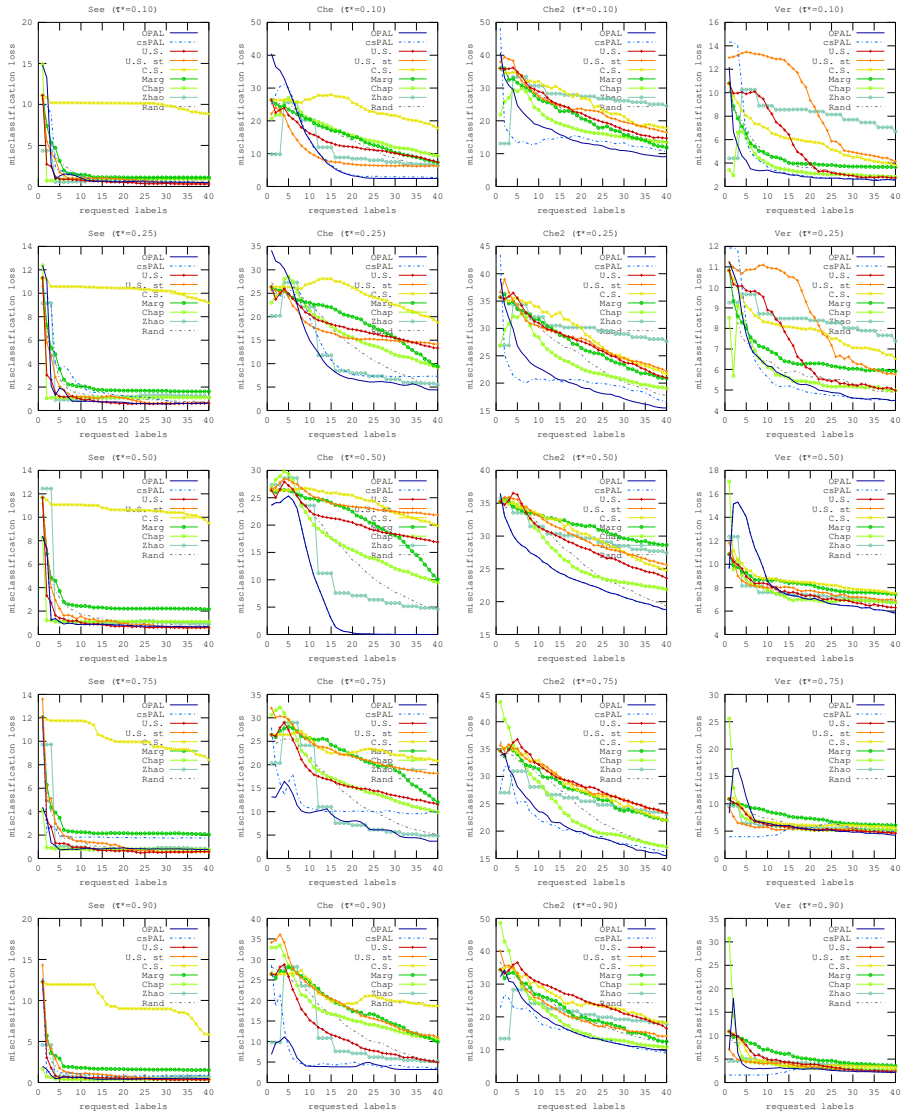


Fig. 5 Misclassification loss curves for presented algorithms on all data sets with different evaluation cost-ratios (τ^*); early convergence to very low values is best.

for moderately unequal cost-ratios ($\tau = 0.25$ or $\tau = 0.75$). However, this is in accordance to the theoretical observations made in Section 3.2.2, where a strongest effect for moderately unequal misclassification cost ratios was predicted.

Furthermore, we observe that csPAL converges faster (see e.g. Table 3 for 10 label acquisitions). However, this again matches with the theoretical discussion in Section 3.2.2, because we set the budget initially to $m = 40$ (and not to 10), thus OPAL has optimised its learning path for 40 label acquisitions. Evaluating its performance after less steps is therefore slightly malicious. However, the results

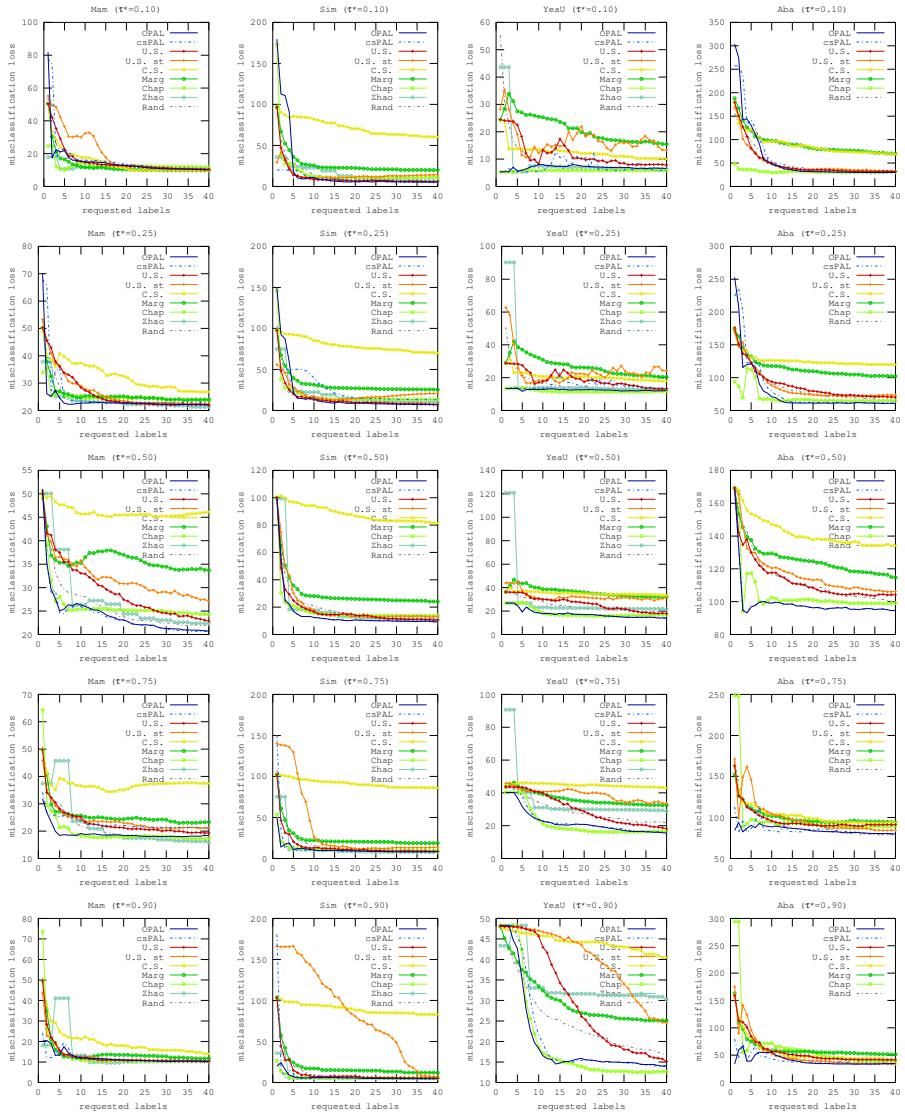


Fig. 6 Continuation of Figure 5 on additional data sets.

indicate that 1) setting the budget correctly to the remaining one is beneficial for final classification performance, and 2) a faster learning is achievable by setting m to low values, if one is willing to forfeit long-term performance.

(2) OPAL's performance is superior
Measuring the overall performance of active learning methods over different data sets and cost-ratios is complex, due to weighting and measuring the characteristics of learning curves, which ideally converge fast to a low final misclassification loss level. Therefore, Tab. 3 provides a summary of the total percentage of wins of

OPAL against each other approach. The learning curves show that OPAL is better than U.S. (with and without self-training) after 6 label requests (when learning becomes meaningful) in most cases. Explanations according to [32, p. 19-20] are that US a) ignores the extend of exploration in a neighbourhood, b) relies on a hypothesis biased by its sampling, and c) is fairly myopic. We can not confirm a superiority of C.S. [10] compared to any other (even random) active learning approach in our experiments. The cost-sensitive error reduction method Marg performs worse than expected. It is outperformed by its cost-insensitive counterpart Chap, maybe due to solely using labelled instances for evaluation, as opposed to the self-labelling used by Chap. Chap and the non-myopic, cost-**in**sensitive error reduction method Zhao sometimes achieve competitive results (esp. on `YeaU`), but only at very high computational costs, which prevented them to be completed on `Aba`. Using the numbers of Tab. 3, Rand is surprisingly the best competitor. All in all, we can argue that OPAL outperforms all other tested algorithms with high significance (see Tab. 3) and has a good trade-off between fast convergence and low final misclassification loss value. Although such an evaluation is not in the scope of this paper, the results on `YeaU` indicate that OPAL is also suitable for unbalanced data sets.

(3) OPAL's runtime in comparison with training set size
To experimentally verify OPAL's time complexity (cmp. Section 3.2.1), we measured the time in seconds per run used by each active learning process and summed it over all 40 label acquisitions in Table 4 (all differences w.r.t. OPAL are significant at level 0.001). Obviously, Rand is fastest, followed by U.S., C.S., csPAL and OPAL, which slow down constantly with increasing training set size. Our myopic approach csPAL is just slightly slower than U.S., due to its more complex value calculation. OPAL takes about 7 times longer than csPAL, because it computes the G_{OPAL} more often when searching for the optimal budget over $m = 1, 2, \cdots, 40$. In contrast, the execution times of Marg, Chap, and Zhao explode on bigger data sets, taking for a single cost ratio more than 8 (Marg), 13 (Chap), or 84 (Zhao) hours. Their calculations on Aba were aborted after one week.

5 Conclusion

In this paper, we addressed the problem of fast, non-myopic active learning for binary classification in cost-sensitive applications. In such applications, unlabelled data is abundant but annotation capacities are limited and require an efficient allocation between labelling candidates. Furthermore, the costs of misclassifications differ between classes, and ultimately the optimal candidate given a remaining labelling budget should be chosen.

We proposed a novel approach, OPAL, that optimises probabilistic active learning for such situations. Given the misclassification cost ratio and remaining budget, which are predetermined by the application, our approach follows a smoothness assumption and computes the expected misclassification loss reduction within a candidate's neighbourhood. For this expectation over the true posterior in the neighbourhood and over the subsequent label realisations therein, we derived a fast, closed-form solution. This allows to select the candidate that reduces the expected misclassification loss in its neighbourhood the most. We have shown

that for a myopic setting, our approach runs in asymptotically linear time in the size of the candidate pool. For the non-myopic setting, we have shown that an additional factor that is solely $O(m \cdot \log m)$ in the budget size is required. Furthermore, we have illustrated the effect of the non-myopic extension, indicating its usefulness for unequal misclassification costs. This is confirmed in experimental evaluations on several synthetic and real-world data sets, where our approach has shown comparable or better classification performance than several uncertainty sampling- or error-reduction-based active learning strategies, both in cost-sensitive and cost-insensitive settings. Our fast approach requires no tunable parameters, yet it is simple to implement, and it neither requires an evaluation sample, nor self-labelling. Thus, its natural extension to data streams has not missed our attention. However, this remains to be done in further research.

Acknowledgements We would like to thank our colleagues from the University Magdeburg and the KMD lab, in particular Myra Spiliopoulou, Pawel Matuszyk and Christian Braune. Furthermore, we thank the anonymous reviewers for their helpful comments and suggestions.

References

1. Asuncion, A., Newman, D.J.: UCI machine learning repository (2013). URL `http://archive.ics.uci.edu/ml/`
2. Attenberg, J., Ertekin, S.: Imbalanced Learning: Foundations, Algorithms, and Applications, chap. Class Imbalance and Active Learning, pp. 101–150. IEEE (2013)
3. Chapelle, O.: Active learning for parzen window classifier. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pp. 49–56 (2005)
4. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-supervised Learning. MIT Press (2006)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal Artificial Intelligence Research (JAIR) **16**, 321–357 (2002)
6. Cohn, D.: Active learning. In: C. Sammut, G.I. Webb (eds.) Encyclopedia of Machine Learning, pp. 10–14. Springer (2010)
7. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. Journal of Artificial Intelligence Research **4**, 129–145 (1996)
8. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: U.M. Fayyad, S. Chaudhuri, D. Madigan (eds.) Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, San Diego, CA, USA, August 15-18, 1999, pp. 155–164. ACM (1999)
9. Elkan, C.: The foundations of cost-sensitive learning. In: B. Nebel (ed.) Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001, pp. 973–978. Morgan Kaufmann (2001)
10. Ferdowsi, Z., Ghani, R., Kumar, M.: An online strategy for safe active learning. In: ICML Workshop on Combining Learning Strategies to Reduce Label Cost (2011)
11. Freytag, A., Rodner, E., Bodesheim, P., Denzler, J.: Labeling examples that matter: Relevance-based active learning with gaussian processes. In: German Conference on Computer Vision (GCPR), pp. 282–291 (2013)
12. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. Knowledge and Information Systems **35**(2), 249–283 (2012)
13. Gantz, J., Reinsel, D.: The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east (2012). URL `http://estonia.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf`
14. Garnett, R., Krishnamurthy, Y., Xiong, X., Schneider, J.G., Mann, R.: Bayesian optimal active search and surveying. In: Proceedings of the 29th Int. Conf. on Machine Learning (ICML 2012). icml.cc / Omnipress (2012)
15. Gopalkrishnan, V., Steier, D., Lewis, H., Guszcza, J.: Big data, big business: Bridging the gap. In: Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, BigMine '12, pp. 7–11. ACM, New York, NY, USA (2012)

16. Hand, D.J.: Measuring classifier performance: a coherent alternative to the area under the roc curve. Machine Learning **77**(1), 103–123 (2009)
17. He, H., Ma, Y. (eds.): Imbalanced Learning: Foundations, Algorithms, and Applications. IEEE (2013)
18. Krempl, G., Kottke, D., Spiliopoulou, M.: Probabilistic active learning: A short proposition. In: Proceedings of the 21st European Conf. on Artificial Intelligence (ECAI2014), August 18 – 22, 2014, Prague. IOS Press (2014)
19. Krempl, G., Kottke, D., Spiliopoulou, M.: Probabilistic active learning: Towards combining versatility, optimality and efficiency. In: Proceedings of the 17th Int. Conf. on Discovery Science (DS), Bled, Lecture Notes in Computer Science. Springer (2014)
20. Krempl, G., Zliobaitė, I., Brzeziński, D., Hüllermeier, E., Last, M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopoulou, M., Stefanowski, J.: Open challenges for data stream mining research. SIGKDD Explorations (2014). Special Issue on Big Data (to appear)
21. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '94, pp. 3–12. Springer-Verlag New York, Inc., New York, NY, USA (1994)
22. Liu, A., Jun, G., Ghosh, J.: A self-training approach to cost sensitive uncertainty sampling. Machine Learning **76**, 257–270 (2009)
23. Liu, A., Jun, G., Ghosh, J.: Spatially cost-sensitive active learning. In: Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA, pp. 814–825. SIAM (2009)
24. Liu, A.Y.c.: Active learning in cost-sensitive environments. Ph.D. thesis, University of Texas, Electrical and Computer Engineering (2009)
25. Margineantu, D.D.: Active cost-sensitive learning. In: Proceedings of the 19th international joint conference on Artificial intelligence, IJCAI'05, pp. 1622–1623. Morgan Kaufmann Publishers Inc. (2005)
26. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: Advances in Neural Information Processing Systems 14, NIPS (2001)
27. Parker, C.: An analysis of performance measures for binary classifiers. In: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM2011), pp. 517 – 526. IEEE (2011)
28. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in Fortran 77: The Art of Scientific Computing, 2 edn. Cambridge University Press (1992)
29. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proceedings of the 18th Int. Conference on Machine Learning, ICML 2001, Williamstown, MA, USA, ICML '01, pp. 441–448. Morgan Kaufmann Publishers Inc. (2001)
30. Schein, A.I., Ungar, L.H.: Active learning for logistic regression: an evaluation. Machine Learning **68**(3), 235–265 (2007)
31. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, Madison, Wisconsin, USA (2009). URL `http://pages.cs.wisc.edu/~bsettles/pub/settles.activelearning.pdf`
32. Settles, B.: Active Learning. No. 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers (2012)
33. Tomanek, K., Hahn, U.: Reducing class imbalance during active learning for named entity annotation. In: Y. Gil, N. Fridman Noy (eds.) Proceedings of the 5th International Conference on Knowledge Capture (K-CAP 2009), September 1-4, 2009, Redondo Beach, California, USA, pp. 105–112. ACM (2009)
34. Vijayanarasimhan, S., Jain, P., Grauman, K.: Far-sighted active learning on a budget for image and video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 13-18 June 2010, San Francisco, CA, pp. 3035 – 3042. IEEE (2010)
35. Zhao, Y., Yang, G., Xu, X., Ji, Q.: A near-optimal non-myopic active learning method. In: Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012, pp. 1715–1718. IEEE (2012)
36. Zhu, J., Wang, H., Tsou, B.K., Ma, M.Y.: Active learning with sampling by uncertainty and density for data annotations. IEEE Transactions on Audio, Speech & Language Processing **18**(6), 1323–1331 (2010)
37. Zliobaitė, I., Bifet, A., Pfahringer, B., Holmes, G.: Active learning with drifting streaming data. IEEE Transactions on Neural Networks and Learning Systems **PP**(99) (2013)

6 Appendix

6.1 Bandwidth (σ) Tuning

The plots in Fig. 7 below show the misclassification loss (y-axis) on the test set, given full label information on the training sets of size 40, for different σ-bandwidths of the Parzen window classifier. The maximal bandwidth (leftmost values of the x-axis) corresponds to a non-discriminating classifier, which simply classifies any instance into the class with higher misclassification cost. Thus, ideally each cost-ratio/data-set-combination exhibits a unique minimum that is smaller than this rightmost value. Combinations with monotonically decreasing misclassification loss curves indicate ill-posed learning problems.

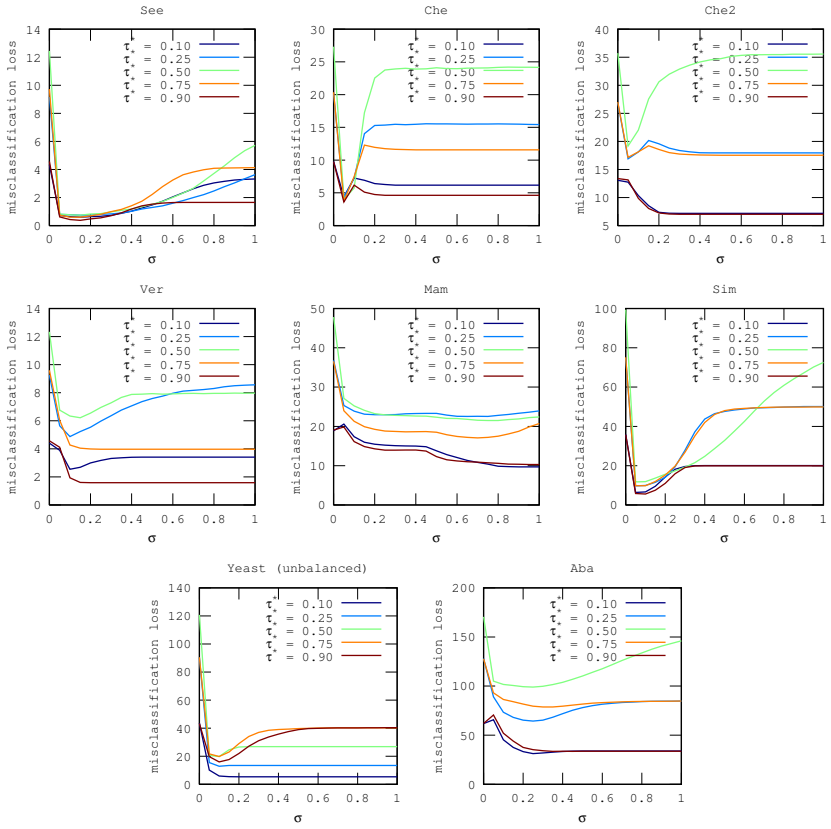


Fig. 7 Misclassification loss for different bandwidth (σ) values of a Parzen window classifier

6.2 Cost-Sensitivity (cont.)

In Figure 8, we continue the results from Fig. 4 on the relevance of the cost-sensitiveness, for details see Section 4.2.

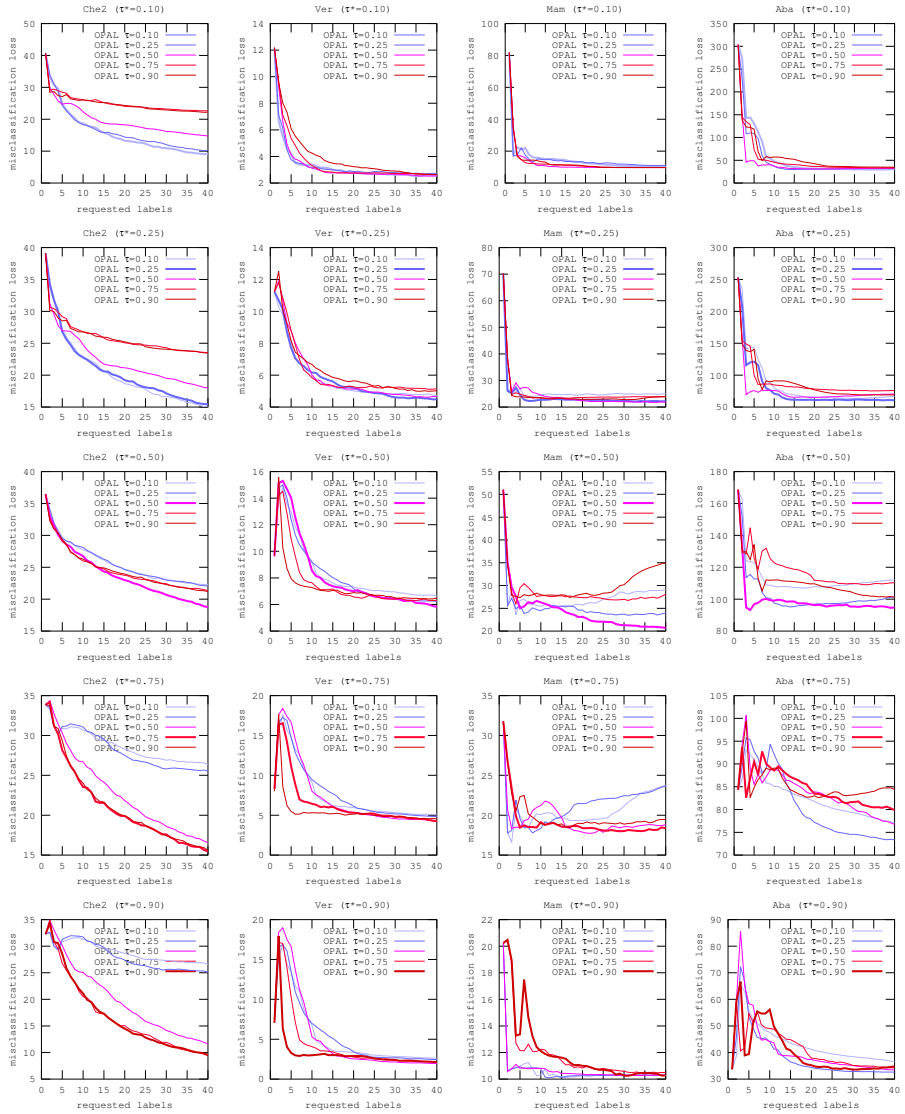


Fig. 8 Misclassification loss curves for OPAL on a selection of data sets with different cost-ratios (τ) for active learning and true cost-ratios (τ^*) for evaluation; curves for correct cost-ratios ($\tau = \tau^*$) are plotted in bold and should be superior; early convergence to very high low values is best. Continuation of Fig. 4 on additional data sets.

6.3 Detailed Derivation of Eq. 24 from Eq. 22

Starting with Eq. 22, we apply Eq. 12, the misclassification loss def. from Eq. 16 and expand the latter by the cost-optimal classification rule from Eq. 18:

$$E_{cur} = \int_0^1 \mathrm{Beta}_{\alpha,\beta}(p) \cdot ML_{p,\tau}(\hat{p})\,\mathrm{d}p \quad (37)$$

$$= \int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \cdot q(\tau-p) + p(1-\tau)\,\mathrm{d}p \quad (38)$$

$$= \int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \cdot \begin{cases} 0(\tau-p) + p(1-\tau) & \hat{p} < \tau \\ (1-\tau)(\tau-p) + p(1-\tau) & \hat{p} = \tau \\ 1(\tau-p) + p(1-\tau) & \hat{p} > \tau \end{cases} \mathrm{d}p \quad (39)$$

$$= \int_0^1 \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \cdot \begin{cases} p(1-\tau) & \hat{p} < \tau \\ (1-\tau)\tau & \hat{p} = \tau \\ \tau(1-p) & \hat{p} > \tau \end{cases} \mathrm{d}p \quad (40)$$

Following Eq. 12, we set $\alpha = n\hat{p} + 1$ and $\beta = n(1-\hat{p}) + 1$, and obtain Eq. 24.

6.4 Detailed Derivation of Eq. 33 from Eq. 32

Using the definition of the Beta Integral

$$\int_0^1 x^a (1-x)^b\,\mathrm{d}x = \mathrm{Beta}(a+1, b+1) = \frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+2)} \quad (41)$$

and setting

$$a = \begin{cases} n\hat{p} + k + 1 & \frac{n\hat{p}+k}{n+m} < \tau \\ n\hat{p} + k & \frac{n\hat{p}+k}{n+m} \geq \tau \end{cases} \quad (42)$$

$$b = \begin{cases} n + m - n\hat{p} - k & \frac{n\hat{p}+k}{n+m} \leq \tau \\ n + m - n\hat{p} - k + 1 & \frac{n\hat{p}+k}{n+m} > \tau \end{cases} \quad (43)$$

we can express the first factors in the integral in Eq. 32 and thus derive Eq. 33:

$$\mathrm{I}_{ML}(n, \hat{p}, \tau, m, k) = \quad (44)$$

$$= \binom{m}{k} \cdot \int_0^1 p^{n\cdot\hat{p}+k} \cdot (1-p)^{n+m-n\cdot\hat{p}-k} \cdot \begin{cases} p \cdot (1-\tau)dp & \frac{n\hat{p}+k}{n+m} < \tau \\ (\tau - \tau^2)dp & \frac{n\hat{p}+k}{n+m} = \tau \\ \tau \cdot (1-p)dp & \frac{n\hat{p}+k}{n+m} > \tau \end{cases} \quad (45)$$

$$= \binom{m}{k} \cdot \begin{cases} (1-\tau) \cdot \frac{\Gamma(1-k+m+n-n\hat{p})\Gamma(2+k+n\hat{p})}{\Gamma(3+m+n)} & \frac{n\hat{p}+k}{n+m} < \tau \\ (\tau-\tau^2) \cdot \frac{\Gamma(1-k+m+n-n\hat{p})\Gamma(1+k+n\hat{p})}{\Gamma(2+m+n)} & \frac{n\hat{p}+k}{n+m} = \tau \\ \tau \cdot \frac{\Gamma(2-k+m+n-n\hat{p})\Gamma(1+k+n\hat{p})}{\Gamma(3+m+n)} & \frac{n\hat{p}+k}{n+m} > \tau \end{cases} \quad (46)$$