This is an author manuscript of the paper:

Georg Krempl, Daniel Kottke, Myra Spiliopoulou.
Probabilistic Active Learning: A Short Proposition.
Proceedings of the 21st European Conference on Artificial Intelligence (ECAI2014), August 18 – 22, 2014, Prague, Czech Republic.
Published by IOS Press, available at: http://www.iospress.nl/book/ecai-2014/

For a companion website to this paper, please consult: http://kmd.cs.ovgu.de/res/pal/

Probabilistic Active Learning: A Short Proposition

Georg Krempl[1] and **Daniel Kottke**[1] and **Myra Spiliopoulou**[1]

Abstract. Active Mining of Big Data requires fast approaches that ideally select for a user-specified performance measure and arbitrary classifier the optimal instance for improving the classification performance. Existing generic approaches are either slow, like error reduction, or heuristics, like uncertainty sampling. We propose a novel, fast yet versatile approach that directly optimises any user-specified performance measure: Probabilistic Active Learning (PAL).

PAL follows a smoothness assumption and models for a candidate instance both the true posterior in its neighbourhood and its label as random variables. By computing for each candidate its expected gain in classification performance over both variables, PAL selects the candidate for labelling that is optimal in expectation. PAL shows comparable or better classification performance than error reduction and uncertainty sampling, has the same asymptotic linear time complexity as uncertainty sampling, and is faster than error reduction.

1 INTRODUCTION

In some applications of machine learning to large data pools and fast data streams, features are cheap but labels are costly, for example due to human annotation efforts [5]. This motivates *active learning* (AL) [8] approaches that actively select the instance, which –once incorporated into the training set– will yield the highest gain in terms of a classification performance measure. Ideally, such an approach allows a) optimisation of an arbitrary, user-defined performance measure, b) is fast and scalable , and c) is usable with any classifier technology.

Each of the existing approaches offers some of the above qualities, but not a *combination* of them *in a single* approach. We propose a novel, *probabilistic active learning (PAL)* approach[2] that fills this gap. As expected error reduction (ER) [7], PAL is not limited to a particular classifier technology or performance measure. Like fast uncertainty sampling (US) [6], PAL requires only linear asymptotic time for selecting the best instance from a pool of labelling candidates. We will present PAL in the next section 2, before relating it to existing approaches in section 3 and evaluating it in section 4.

2 PROBABILISTIC ACTIVE LEARNING

We address the *pool-based* [9] active learning scenario for binary classifiers, where an active classifier has access to a pool of unlabelled instances $\mathcal{U} = \{(x, .)\}$. Repeatedly, the best instance $(x^*, .) \in \mathcal{U}$ is selected, its label y^* is requested from an oracle, and it is moved from $\mathcal{U}$ to the set of labelled instanced $\mathcal{L}$.

Following the common smoothness assumption [3], we consider that an instance x influences the classification the most in its neighbourhood. Thus, the impact of an additional label primarily depends on the already obtained labels in its neighbourhood. We summarise these by their *absolute* number n, and the share of positives $\hat{p}$ therein, yielding the *label statistics* $\text{ls} = (n, \hat{p})$. Here, n is obtained by counting the similar labelled instances for pre-clustered or categorical data, or approximated by frequency estimates such as kernel frequency estimates for smooth, continuous data. Thus, in x's neighbourhood, n expresses the absolute quantity of labelled information, whereas the density d_x of unlabelled instances quantifies the importance of this neighbourhood, i.e. the share of future classifications that will take place therein compared to other regions of the feature space.

Given a candidate instance $(x, .)$ with ls and d_x, we want to compute the *overall gain* in classification performance if requesting its label. This gain depends also on the realisation of the candidate's label y, and of the true posterior probability p of the positive class within the neighbourhood. Both values are unknown, thus we use a probabilistic approach and model the candidate's label Y and the true posterior of the positive class P as random variables. This allows to compute the *expected value* of the gain in performance over all different true posteriors and label realisations, which we denote as *probabilistic gain*[3] (pgain). Weighting the latter with d_x, we obtain an estimate on the impact of x's label on the overall classification performance. Subsequently, we select among all instances the one with highest density-weighted probabilistic gain.

The figure below summarises PAL's pseudo-code. Iterating over the candidate pool $\mathcal{U}$ (lines 2-6), for each candidate x one computes its label statistics $\text{ls}_x = (n_x, \hat{p}_x)$, its density weight d_x, and its probabilistic gain by using numerical integration, which is then weighted by its density weight to obtain g_x. Finally, the candidate with the highest density-weighted probabilistic gain is selected (line 7).

```
1: function PoolBasedPAL(U,L)
2:     for x ∈ U do
3:         (n_x, p̂_x) ← labelstatistics(x, L)
4:         d_x ← densityweight(x, L ∪ U)
5:         g_x ← pgain((n_x, p̂_x)) · d_x
6:     end for
7:     return x* ← arg max_{x∈U}(g_x)
8: end function
```

We propose to precompute d_x, as $\mathcal{U} \cup \mathcal{L}$ is static, and to use probabilistic classifiers to compute the absolute frequency estimates needed for ls. Thus, lines 3–4 are constant-time operations, but the probabilistic gain (pgain) computation deserves further discussion:

$$\text{pgain}(\text{ls}) = \mathtt{E}_p\Big[\mathtt{E}_y\left[\text{gain}_p(\text{ls}, y)\right]\Big] \tag{1}$$

$$= \int_0^1 \text{Beta}_{\alpha,\beta}(p) \cdot \sum_{y\in\{0,1\}} \text{Ber}_p(y) \cdot \text{gain}_p(\text{ls}, y)\,\mathrm{d}p \tag{2}$$

Here, $\text{gain}_p(\text{ls}, y)$ is the candidate's $(x, .)$ performance gain given its label realisation y and the neighbourhood's true posterior p:

$$\text{gain}_p(\text{ls}, y) = \text{perf}_p\left(\frac{n\hat{p} + y}{n + 1}\right) - \text{perf}_p(\hat{p}) \tag{3}$$

$\text{perf}_p(\hat{p})$ is an arbitrary point-performance measure (e.g. accuracy), indicating the classification performance within the neighbourhood, given the true posterior p and a posterior estimate $\hat{p}$ by the classifier.

[1] Knowledge Management & Discovery Lab, Univ. Magdeburg, Germany, email: [georg.krempl|daniel.kottke|myra]@iti.cs.uni-magdeburg.de

[2] **See the companion website: `http://kmd.cs.ovgu.de/res/pal/`**

[3] We do this to differentiate it from the expected gain as in expected error reduction methods like [2], where expectation is solely over label outcomes.

$\mathrm{Ber}_p(y)$ is the probability of the Bernoulli-distributed random variable Y producing the label realisation $y \in \{0, 1\}$ (1 corresponding to a positive label), whose parameter p corresponds to the true posterior, which itself is the realisation of the Beta-distributed random variable P with parameters $\alpha = n \cdot \hat{p} + 1$ and $\beta = n \cdot (1 - \hat{p}) + 1$ and the resulting probability density function $\mathrm{Beta}_{\alpha,\beta}(p)$. Note that this Beta-distribution and its particular parameters are the result of using a Bayesian approach that assumes a uniform prior $g(p)$ for the true posterior probability and computes the normalised likelihood $\omega_{\mathcal{L}}(p)$ of p given the data in $\mathcal{L}$, that is:

$$\omega_{\mathcal{L}}(p) = \frac{L(p|\mathcal{L})g(p)}{\int_0^1 L(\psi|\mathcal{L})g(\psi)\mathrm{d}\psi} = \mathrm{Beta}_{\alpha,\beta}(p) \tag{4}$$

Thus the parameters α and β of the normalised likelihood correspond to the absolute numbers of positive and negative labels (plus one).

3 DISCUSSION AND RELATED WORK

Our approach is related to expected error reduction (ER), first proposed by [4], where for each labelling candidate the expected reduction in classification error is computed. While in [4] closed-form solutions are derived for optimal data selection for two specific learning methods, [7] proposed a generic ER approach, both with respect to arbitrary performance measures and classifiers: using a Monte Carlo sampling approach, it estimates the performance on a labelled validation sample $\mathcal{V}$, rather than integrating over the full feature distribution $\mathcal{P}r(x)$ as in [4]. Furthermore, it uses the posterior estimate $\hat{p} = \hat{\mathcal{P}}r(y|x)$ provided by the current classifier as proxy for the true posterior $\mathcal{P}r(y|x)$ that is required for the expectation over the label realisations y. However, [2] noted that this proxy is not reliable if solely few labels are available (as common in active learning) and requires regularisation approaches such as using Beta priors.

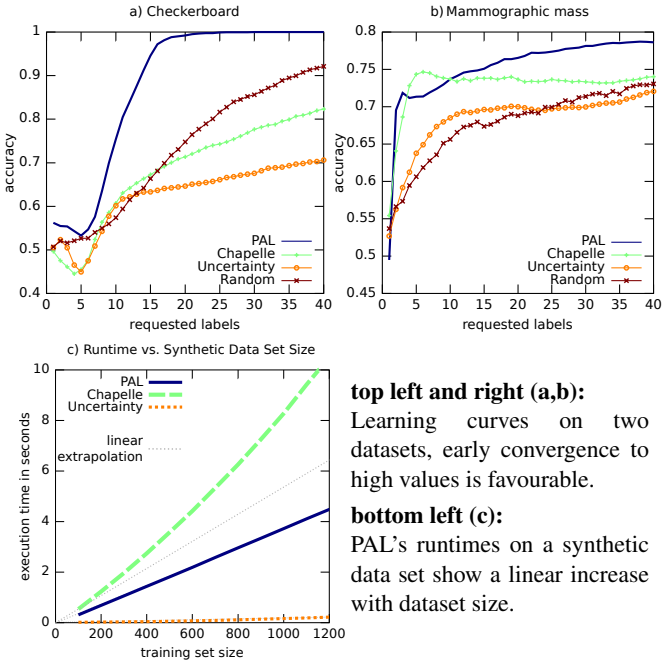
In contrast to ER, expectation in PAL is also over the true posterior p, and evaluation is done using the label statistics within an instances neighbourhood, rather than simulating classifier updates and evaluating them on a validation sample. The latter makes ER prohibitively slow [8], as even for incremental classifiers its asymptotic time complexity is $O(|\mathcal{V}| \cdot |\mathcal{U}|)$. PAL's time complexity is $O(|\mathcal{U}| \cdot q \cdot 2) = O(|\mathcal{U}|)$, as the probabilistic gain computation for each candidate in $\mathcal{U}$ according to eq. 2 requires a constant number of q numerical integration steps ($q = 50$ was used in our experiments), and summarising over the two potential label outcomes $\{0, 1\}$.

This is identical to the asymptotic time complexity of uncertainty sampling (US), proposed in [6]. US uses simple uncertainty measures [9], like sample margin, confidence, or entropy as proxies for a candidate's value, and selects the candidate with maximal uncertainty. However, these proxies do not consider the number of similar instances, neither does US directly optimise a performance measure.

4 EXPERIMENTAL EVALUATION

We compare PAL to the error reduction approach proposed in [2], to uncertainty sampling proposed in [6] (using confidence [9]), and to random sampling. We use the synthetic Checkerboard dataset from [2], the Mammographic mass dataset from [1], and a synthetic dataset consisting of a Gaussian mixture model in 2d with varying training set sizes for speed-testing. For comparison with [2], we use a Parzen Window classifier with pre-tuned bandwidth (0.1, 0.1, and 0.08, resp.). Evaluation was done by averaging the performance over 100 randomly generated partitionings in training and test subsets.

The results are shown in the figure below, where a) and b) are plots of the approaches' learning curves, and c) is a plot of the execution time relative to the pool size. Overall, PAL yields superior classification performance than all other approaches, while its runtime solely increases linearly in the pool size.



top left and right (a,b): Learning curves on two datasets, early convergence to high values is favourable.

bottom left (c): PAL's runtimes on a synthetic data set show a linear increase with dataset size.

ACKNOWLEDGEMENTS

We thank Vincent Lemaire from Orange Labs, France, for the insightful discussion on this approach.

REFERENCES

[1] Arthur Asuncion and David J. Newman. UCI ML repository, 2013.

[2] Olivier Chapelle, 'Active learning for parzen window classifier', in *Proc. 10th Int. Workshop on AI and Statistics*, pp. 49–56, (2005).

[3] *Semi-Supervised Learning*, eds., Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, MIT Press, 2006.

[4] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan, 'Active learning with statistical models', *J. of AI Research*, **4**, 129–145, (1996).

[5] Vivekanand Gopalkrishnan, David Steier, Harvey Lewis, and James Guszcza, 'Big data, big business: Bridging the gap', in *Proc. 1st Int. Workshop on Big Data, Streams and Heterogeneous Source Mining, Big-Mine 2012*, pp. 7–11. ACM, (2012).

[6] David D. Lewis and William A. Gale, 'A sequential algorithm for training text classifiers', in *Proc. 17th ann. int. ACM SIGIR conf. on research and development in information retrieval*, pp. 3–12, (1994).

[7] Nicholas Roy and Andrew McCallum, 'Toward optimal active learning through sampling estimation of error reduction', in *Proc. 18th Int. Conf. on ML, ICML 2001*, pp. 441–448. Morgan Kaufmann, (2001).

[8] Burr Settles, 'Active learning literature survey', Computer Sciences Technical Report 1648, University of Wisconsin, USA, (2009).

[9] Burr Settles, *Active Learning*, number 18 in Synthesis Lectures on AI and ML, Morgan and Claypool Publishers, 2012.