Adaptive Active Learning with Ensemble of Learners and Multiclass Problems

Wojciech Marian Czarnecki

Faculty of Mathematics and Computer Science,
Jagiellonian University, Cracow, Poland
wojciech.czarnecki@uj.edu.pl
http://czarnecki.gmum.net

Abstract. Active Learning (AL) is an emerging field of machine learning focusing on creating a closed loop of learner (statistical model) and oracle (expert able to label examples) in order to exploit the vast amounts of accessible unlabeled datasets in the most effective way from the classification point of view.
This paper analyzes the problem of multiclass active learning methods and proposes to approach it in a new way through substitution of the original concept of predefined utility function with an ensemble of learners. As opposed to known ensemble methods in AL, where learners vote for a particular example here we use them as a black box mechanisms for which we try to model the current competence value using adaptive training scheme.
We show that modeling this problem as Multi-Armed Bandit problem and applying even very basic strategies bring significant improvement to the AL process.

Keywords: active learning, ensemble, classification, multiclass

1 Introduction

Classical supervised machine learning methods require big labeled datasets to construct good models. Unfortunately, in real life applications it is often the case that, while large amounts of data may be available, a significant portion of them misses the true labeling. Obtaining such information commonly requires substantial amounts of time/costs (like labeling video recordings, tagging text corpora or synthesis and testing of new kinds of drugs). *Active learning* [24] addresses this issue by introducing the label querying step into the model's training process to minimize the total cost of building the most accurate classifier.

To the authors' best knowledge, not much work has addressed the problem of dealing with active learning ensembles. There has been many approaches to build active learning strategies on the ensembles of models [25] or extreme hypotheses [13] but they all operate on the different level of abstraction.

In this paper we present a method of creating a complex active learning strategy from an ensemble of existing strategies. We model the problem of adaptively selecting the best ones as a well known Multi-Armed Bandit problem, which is used as a model in many fields of computer science [12] and machine learning [7]. We analyze the generic scheme of such approach and define few families of possible reward functions. Then we focus on applicability of particular selection strategies in this context.

2 Related work

This work is a continuation of the short communicate entitled "Adaptive Active Learning as the Multi Armed Bandit Problem" [10]. Results from this paper significantly extend previous ones through more theoretical analysis of the process, deeper analysis of empirical behaviour, extension to multiclass problems and more valuable reward function.

There are many active learning strategies working with ensembles of learners in the sense of classifiers. Approaches such as Query by Disagreement (QBD [13]) or Query by Committee (QBC [25]) assume that there is one, fixed utility function which analyzes multiple classifiers (elements of an ensemble) and a meta-strategy which makes a consensus decision (such as majority voting or entropy). In this work we analyze a problem one layer of abstraction higher, with learner we shall denote the actual utility function which makes "small decisions". So we are working in the scenario where we have multiple utility functions, working on a single machine learning model (which can be an ensemble itself, in fact we perform evaluation using entropy QBD querying as an element of a learners ensemble).

3 Adaptive Active Learner

In classic Active Learning pool based approach [26, 27] we have a set of unlabeled data $\mathcal{U}$ and a learner defined by its utility function u such that

$$u : \mathcal{H} \times \mathcal{U} \to \mathbb{R}.$$

In given iteration we have hypothesis (model) h and simply select sample to label it with

$$q = \arg\max_{x \in \mathcal{U}} u(h, x).$$

This means that if we denote by h_i the hypothesis in i'th iteration, by A the training algorithm which given training set returns a hypothesis and an oracle over K-class problem $\boldsymbol{o} : \mathcal{U} \to \{1, ..., K\}$, we obtain:

$$h_{i+1} = \mathrm{A}(\{(q_k, \boldsymbol{o}(q_k))\}_{k=1}^{i}),$$

where

$$q_i = \arg\max_{x \in \mathcal{U}_i} u(h_i, x)$$

$$\mathcal{U}_{i+1} = \mathcal{U}_i \setminus \{q_i\}.$$

If we look at the time dimension of our dynamical learning process it appears that only h depends on time (as classifier learns over time), while u is considered a constant function.

The Adaptive Active Learner (A^{2}L) proposed by Czarnecki et al. [10] assumes that also u itself depends on time and can benefit from past experiences, meaning that

$$q_i = \arg\max_{x \in \mathcal{U}_i} u_i(h_i, x)$$

and there is some adaptive process f such that

$$u_{i+1} = f((h_k)_{k=1}^i, (u_k)_{k=1}^i, (q_k)_{k=1}^i, (\boldsymbol{o}(q_k))_{k=1}^i),$$

where $(a_k)_{k=1}^i$ denotes *sequence* of a_i from $k = 1$ to i.

In general f could be any function returning an utility function so the problem of finding a good one is extremely complex. In the following sections we will focus on a narrowing of space of possible f to the family of functions on some finite set.

3.1 A^{2}L for Ensemble of learners

Let us assume that we are given a finite set of learners, denoted by their corresponding utility functions

$$L = \{u^{(1)}, ..., u^{(K)}\}$$

and that in each iteration we simply choose one of these learners to query, using some previous knowledge on their competences in current task and timeframe. One of possible statistical models which can help us perform adaptation over time is Multi-Armed Bandit problem [23]. Given some finite set of processes (bandits) we iteratively sample (play) them and receive a value (reward). The aim of this model is to maximize the cumulative reward over time having no prior knowledge regarding particular bandits. We only observe results of playing at given machine. The problem of finding best strategy can be stated as a minimization of so called regret $\mathcal{R}$, defined as a difference between sum of gained rewards and the sum of rewards obtained by some hypothetical optimal selection.

$$\underset{s}{\text{minimize}} \;\; \mathcal{R}(s) = \sum_{k=1}^{T} r_k^{opt} - \sum_{k=1}^{T} r_k^s \iff \underset{s}{\text{maximize}} \;\; \sum_{k=1}^{T} r_k^s,$$

where r_k^s is the reward obtained in kth iteration using strategy s and *opt* is the optimal strategy.

In order to model A^{2}L using MAB as in [10] we consider learners as machines and querying them as playing. The only thing missing is some reward function r which we want to optimize and we can plug in any existing MAB algorithm (such as ε-greedy [16], UCB [3], EXP3 [4], ...) as f.

$$f((h_k)_{k=1}^i, (u_k)_{k=1}^i, (q_k)_{k=1}^i, (\boldsymbol{o}(q_k))_{k=1}^i) := \text{MAB}((u_k)_{k=1}^i, (r_k)_{k=1}^i),$$

where

$$r_i = r((h_k)_{k=1}^i, (u_k)_{k=1}^i, (q_k)_{k=1}^i, (\boldsymbol{o}(q_k))_{k=1}^i).$$

One of the simplest reward functions one can define is a characteristic function of predicting an incorrect label

$$r((h_k)_{k=1}^i, (u_k)_{k=1}^i, (q_k)_{k=1}^i, (\boldsymbol{o}(q_k))_{k=1}^i) = r_{0/1}(h_i, q_i, \boldsymbol{o}(q_i)) = 1_{h_i(q_i) \neq \boldsymbol{o}(q_i)}.$$

Using such reward function we prefer strategies querying points which are currently incorrectly labeled by our classifier. As a result we maximize accuracy of the underling machine learning model.

Theorem 1. *Given A^2L with UCB1 strategy, set of active learners L and $r_{0/1}$ reward function, assuming that given learner's reward in each iteration is stochastic and independent on process history the maximum number of missed (not queried) mislabeled points which could be found using different selection of learners after T iterations is $\mathcal{O}(\log T)$. In general this bound cannot be improved.*

Proof. This is a direct consequence of the UCB1 MAB bounds on regret function and the fact that $r_{0/1}$ equals 1 if and only if current hypothesis mislabels given point. This bound realizes the theoretical lower bound of MAB error [3] of $\Omega(\log T)$ so in general it cannot be asymptotically improved. □

Slightly more complex reward function is to measure the increase in the generalization capabilities of a particular model under given evaluation metric m on some representative[1] validation set V.

$$r_m(h_i, h_{i-1}; V) = m(h_i(V), \mathbf{o}(V)) - m(h_{i-1}(V), \mathbf{o}(V)).$$

Using such reward function we can get even better theoretical bound on the learning process.

Theorem 2. *Given A^2L with UCB1 strategy, set of active learners L and r_m reward function, assuming that given learner's reward in each iteration is stochastic and independent on process history, the achieved generalization capabilities in terms of metric m is at most $\mathcal{O}(\log T)$ worse than the one achieved by optimal selection of learners after T iterations. In general this bound cannot be improved.*

Proof. This is a direct consequence of the UCB1 MAB bounds on reward function and the fact that

$$\begin{aligned}\sum_{k=1}^{T} r_{m_k} &= \sum_{k=1}^{T} [m(h_k(V), \mathbf{o}(V)) - m(h_{k-1}(V), \mathbf{o}(V))] \\ &= m(h_T(V), \mathbf{o}(V)) - m(h_0(V), \mathbf{o}(V))\end{aligned}$$

[1] in Practical Considerations section we shortly discuss how to get such set V

is the generalization capability of a given model, dependent only on the final hypothesis h_T and the initial hypothesis h_0. Optimality is analogous to the one from Theorem 1. □

3.2 Multi armed bandit algorithms

In this section we briefly summarize Multi Armed Bandits (MAB) algorithms used in this work. According to extensive evaluation on both artificial and real datasets [16] it has been shown that ε-greedy and UCB, very simple methods of MAB perform well in reality. We use following notation to denote mean reward of jth learner

$$\bar{r}_j = \sum_{k:P_j} \tfrac{r_k}{|P_j|},$$

where $P_j = \{k : u_k = u^{(j)}\}$ is a set of iterations' indices where jth learner was used.

ε-greedy With probability $1 - \varepsilon$ we select learner which maximized the mean rewards obtained till this iteration, so we choose u_b such that

$$b = \operatorname*{arg\,max}_{j \in \{1,\ldots,K\}} \bar{r}_j.$$

And with probability ε we just select random b from uniform distribution over learners. In other words for most of the time (assuming that $\varepsilon < 0.5$) we use the most promising learner so far, and for the rest of time we randomly explore other strategies. This is a very greedy approach which might lead to highly suboptimal results if learners competences change over time.

Upper Confidence Bound (UCB1) proposed by Auer, Cesa-Bianchi and Fisher [3] as an elegant approach to the idea of Lai and Robbins [17]. It has been shown that such approach achieves the minimal regret bound of $\Omega(\log T)$ if we assume constant variance of each machine.

$$b = \operatorname*{arg\,max}_{j \in \{1,\ldots,K\}} \left(\bar{r}_j + \sqrt{\tfrac{2 \log T}{|P_j|}} \right).$$

This method achieves surprisingly good results in numerous applications, including its extension to the game trees structure (Upper Confidence Bounds applied to Trees, UCT) which is currently one of the best AI methods for GO [12], beating even deep convolutional networks [8].

Upper Confidence Bound Tuned (UCB1 Tuned) according to authors [3] including the machine's variance helps in determining the most promising exploration candidates.

$$b = \operatorname*{arg\,max}_{j \in \{1,\ldots,K\}} \left(\bar{r}_j + \sqrt{\tfrac{\log T}{|P_j|} \min\left(\tfrac{1}{4}, \operatorname*{var}_{k \in P_j}(r_k) + \sqrt{\tfrac{2 \log T}{|P_j|}} \right)} \right),$$

however to our best knowledge this method lacks many success stories [16].

4 Practical considerations

Proposed approach, as well as most of the existing utility functions, is independent on the choice of underlying classifier. However, from the practical point of view one should use a model which:

- supports multiclass classification, preferably directly[2],
- is able to return a confidence $P(y|x)$ of classification instead of just a label,
- is capable of assessing its own generalization capabilities without creating hold out set never-used during training process.

While two first elements are satisfied by nearly any modern classifier including neural networks [20], true multiclass Support Vector Machines [9] or Extreme Learning Machines [14]. However, last element is much more complex, and requires classifier to be based on bagging [5] or other ensemble technique which allows model to use, at the same time, all data for training and have a set of observations which can be used for unbiased error estimation.

Recent extensive evaluation studies showed that Random Forest [6] is one of the best behaving classifiers available in modern machine learning libraries [11]. This model satisfies all three requirements we listed above and furthermore it works quite well even with very rough estimation of the hyperparameters. This makes it one of the best candidates to use with A^2L (and most other existing active learning strategies).

5 Evaluation

Now we proceed to empirical evaluation of proposed method. Following experiments were performed using code written in Python with use of scikit-learn library [21]. We use three datasets, DIGITS [1], being a collection of 1797 low-resolution (8x8 pixels) hand written digits scans; MNIST [18], the well known dataset of 70,000 hand scans of postal codes and finally CIFAR-10 [15], a dataset of thousands of 32x32 pixels images of various objects from 10 classes. We use Random Forest classifier as a base method, for explanation of this particular choice refer to the Practical Considerations section. Each experiment starts with a small set of randomly selected labeled examples (2 samples from each class for MNIST and DIGITS, and 100 samples from each class for CIFAR-10[3]). For digits recognition we use raw pixels values as the input to the classifier while for images recognition we first perform randomized PCA dimensionality reduction to map $\mathbb{R}^{32\times32}$ to $\mathbb{R}^{20}$.
During these experiments we show three strengths of A^2L, namely:

[2] models such as SVM do not support such operation but using Platt's scaling one can add this type of functionality at the cost of loss of mathematical cohesion of the method

[3] CIFAR-10 is much harder dataset and RF needs more knowledge to start modeling actual concepts

1. ability to exploit diverse characteristics of given set of learners,
2. small vulnerability to the redundant learners,
3. possibility to model completely novel strategies.

Let us focus on the evaluation process. We propose to look at two different tasks which might be the aim of active learning scenario. First, the most natural one is to create a good predictive model using only subset of the training data. We will call this a concept learning, as we are interested in building a classifier which solves the problem for some underlying data densities and so we assume that there are (possibly infinitely many) samples besides our unlabeled pool. The second possibility is to assume that the pool is actually whole input space and we are interested in correct classification of these samples using as small labeling queries as possible. We will refer to this scenario as finite set learning, as we assume that there are no samples besides the pool. Even though these problems might look similar there is at least one fundamental difference between them. The second problem can be solved efficiently through querying the hardest points, as they will be automatically correctly classified,

while leaving the easy samples to our model. In the first scenario this is a much more complex problem as the hardest examples not necessarily are the ones giving most knowledge about the concept being modeled. Both these approaches have their justification and practical applications so we will analyze both scenarios. In order to measure the concept learning capabilities we will use the hold out set to estimate the generalization capabilities which is different than the pool of samples, while for finite set learning we will estimate generalization on all samples from the pool minus queried points. Table 1 summarizes popular learners' utility functions used for multiclass active learning.

method	$u(h, x)$
uncertainty	$-\max_l P(h(x) = l)$
margin	$\lvert \max_l P(h(x) = l) - \max_{l' \neq l} P(h(x) = l')\rvert$
entropy	$-\sum_l P(h(x) = l) \log P(h(x) = l)$
passive	$\mathrm{hash}(x)$

Table 1. Popular utility functions, $P(h(x) = l)$ is models confidence in classifying x as l. By hash we denote perfect hashing function which guarantees random ordering of values.

During experiments on DIGITS dataset all strategies significantly speeded up the learning process (as compared to passive learning, see Table 2) and led to about 86% accuracy in the concept learning scenario and over 95% in finite set learning. One can easily deduce that learners were able to eliminate hard examples from the dataset. Margin based method appeared to perform the best, while entropy learner was much weaker. We applied basic ε-greedy $\mathrm{A}^2\mathrm{L}$ with the learners' ensemble consisting of margin, uncertainty and entropy learners. Not only was this strategy able to deal with existence of weak learner, but achieved results even better than any of the basic methods (despite these approaches yield highly correlated utility functions [24]). This shows some basic resistance to the existence of redundant learners in the ensemble.

DIGITS 200 iterations	Finite Mean Acc	 Final Acc	Concept Mean Acc	 Final Acc
Passive	81.56 ± 0.07	91.90	78.65 ± 0.04	80.91
Uncertainty	86.16 ± 0.09	95.09	81.05 ± 0.04	86.34
Entropy	80.96 ± 0.09	91.74	81.82 ± 0.03	82.74
Margin	86.41 ± 0.10	95.36	82.17 ± 0.04	86.80
A^{2}L $r_{0/1}$ 0.5-greedy {entropy, margin, uncertainty}	86.38 ± 0.10	95.37	$\mathbf{84.08} \pm 0.05$	**87.77**
A^{2}L r_{Acc} UCB1 { margin, uncertainty}	86.91 ± 0.09	95.75	$\mathbf{83.51} \pm 0.04$	**89.06**
A^{2}L r_{Acc} UCB1 $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	85.85 ± 0.09	94.58	$\mathbf{84.95} \pm 0.04$	**89.11**
A^{2}L r_{Acc} UCB1Tuned $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	85.54 ± 0.09	95.73	82.12 ± 0.04	**86.97**

Table 2. Comparison of different strategies on Pen-based DIGITS dataset.

After removal of the entropy learner and change to the more complex strategy (UCB1 with r_{Acc} reward function) we were able to tweak the learning curve and achieve above 89% concept learning ability. This is about 3% increase of the final accuracy after sampling just 200 images as compared to the best learner's result and over 10% better than the passive approach.

During an active learning process some classes might be easy to separate even in the very early stage of the experiment despite the fact that their particular samples still lie in the "uncertain" zone of the input space. As the result uncertainty, margin or entropy sampling will query their labels even though this part of the classification is already completed. In order to deal with this problem we propose to create a family of learners u_i such that they do not query points for which current model returns different label than i. On the rest of the samples they work as simple uncertainty sampling[4].

$$u_i(h, x) = \begin{cases} -\infty & , \text{ if } h(x) \neq i \\ -P(h(x) = i) & , \text{ if } h(x) = i \end{cases}$$

It is easy to notice that such utility function cannot be expressed as single learner (as it would be equivalent of 1 vs all learning with just ith classifier) nor with voting schemes (as each u_i has disjoint competence subspaces meaning

[4] one can analogously define such family for other basic utility functions such as margin or entropy

that $\forall x \exists ! i : u_i(x) \neq -\infty$) so it would degenerate to the uncertainty sampling. However in the A^2L scenario this family gives us ability to select in each iteration which class is now the most important one and sample from it.

MNIST 500 iterations	Finite Mean Acc	Finite Final Acc	Concept Mean Acc	Concept Final Acc
Passive	71.86 ± 0.09	78.50	73.48 ± 0.08	78.89
Uncertainty	66.16 ± 0.07	72.01	70.86 ± 0.07	77.24
Entropy	64.68 ± 0.08	72.09	67.54 ± 0.07	76.66
Margin	68.84 ± 0.16	82.05	71.98 ± 0.17	85.90
A^2L $r_{0/1}$ UCB1 $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	**75.36** ± 0.10	**84.20**	**78.33** ± 0.09	85.82
A^2L r_{Acc} UCB1 $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	**73.32** ± 0.10	81.63	**77.47** ± 0.10	84.10
A^2L r_{Acc} UCB1Tuned $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	**74.33** ± 0.10	**84.27**	**77.75** ± 0.10	**86.02**
A^2L r_{Acc} 0.5-greedy $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	**74.36** ± 0.11	**84.03**	**77.46** ± 0.10	**86.20**

Table 3. Comparison of different strategies on MNIST dataset.

Such learner slightly increases the score on DIGITS dataset (Table 2), however in more complex task of MNIST problem (Table 3) the difference is not that big. One should notice, that this approach is based on uncertainty sampling, and as compared to the non-class based approach it outperforms it by nearly 10% in concept learning and over 12% in finite set learning.

CIFAR-10 350 iterations	Finite Mean Acc	Finite Final Acc	Concept Mean Acc	Concept Final Acc
Passive	26.72 ± 0.02	29.27	24.31 ± 0.02	28.61
Uncertainty	24.81 ± 0.01	27.89	26.20 ± 0.02	27.70
Entropy	25.05 ± 0.02	27.94	23.94 ± 0.02	27.78
Margin	27.28 ± 0.02	29.86	27.97 ± 0.02	30.86
A^2L r_{Acc} UCB1 margin $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	27.13 ± 0.02	29.94	27.12 ± 0.02	30.20
A^2L r_{Acc} UCB1 {margin, uncertainty}	26.99 ± 0.02	**31.07**	28.14 ± 0.02	**31.26**
A^2L r_{Acc} UCB1Tuned margin $\{u_0, u_1, u_2, u_3, \cdots, u_9\}$	**27.88** ± 0.02	**30.31**	**29.09** ± 0.02	**33.13**

Table 4. Comparison of different strategies on CIFAR-10 dataset.

Once we switch to the variance analysis enriched UCB strategy – UCB1Tuned, we get adaptation, which outperforms all basic methods with about 2% in final accuracy on MNIST dataset (in both tasks) and over 5% of mean accuracy, meaning that learning process is much more stable (we achieve good results not only after full 500 iterations but also "in between"). The same method is able to increase final scores also on much harder, CIFAR-10 (Table 4) dataset for about 3% on concept learning and brings a bit weaker effect on the finite set learning.

6 Discussion and conclusions

Multiclass learning is still a challenge for active learning. In binary classification uncertainty sampling is quite successful despite its simplicity. In multiclass approach its natural generalization appears to perform quite weak. Adaptive Active Learning, proposed in this paper, tackles this problem and proposes generic framework in which we can generalize uncertainty sampling to the multiclass scenario. Resulting strategy outperforms (as shown in the evaluation section) state of the art multiclass methods in most cases. Furthermore, proposed approach appears to be able to deal (to some extent) with redundant learners and select the most promising strategies. There seems to be a strong correlation between problems complexity and required multi armed strategy needed to achieve best results.

It remains an open question whether very simple generalization to the multiclass ensembles is the best approach especially when dealing with large classes space (such as hundreds in CIFAR-100 or thousands in Wikipedia). This would cause linear growth of the ensemble size and could be hard to comprehend by MAB methods [16]. One could investigate some grouping/class clustering methods such as those used in HCOC [22] model.

There are also more complex strategies for MAB problems, including Exp family [4], ShiftBand methods [2], as well as a more advanced methods adding feature vectors to each machine such as LinUCB [19]. These methods could be very beneficial in our task where we can associate various strategies statistics such as variance (which helped UCB1 to achieve better scores through UCB1Tuned heuristic) to actually learn the exact relation between these effects.

7 Aknowledgments

The paper was partially funded by National Science Centre Poland Found grant no. 2013/09/N/ST6/03015.

References

1. Alimoglu, F., Alpaydin, E.: Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In: Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96). Citeseer (1996)

2. Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research 3, 397–422 (2003)
3. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning 47(2-3), 235–256 (2002)
4. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: The nonstochastic multiarmed bandit problem. SIAM Journal on Computing 32(1), 48–77 (2003)
5. Breiman, L.: Bagging predictors. Machine learning 24(2), 123–140 (1996)
6. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
7. Carpentier, A., Lazaric, A., Ghavamzadeh, M., Munos, R., Auer, P.: Upper-confidence-bound algorithms for active learning in multi-armed bandits. In: Proceedings of the 22nd international conference on Algorithmic learning theory. pp. 189–203. ALT'11, Springer-Verlag (2011)
8. Clark, C., Storkey, A.: Teaching deep convolutional neural networks to play go. arXiv preprint arXiv:1412.3409 (2014)
9. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. The Journal of Machine Learning Research 2, 265–292 (2002)
10. Czarnecki, W.M., Podolak, I.: Adaptive active learning as a multi-armed bandit problem. Frontiers in Artificial Intelligence and Applications pp. 989–990 (2014)
11. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? The Journal of Machine Learning Research 15(1), 3133–3181 (2014)
12. Gelly, S., Silver, D.: Monte-Carlo tree search and rapid action value estimation in computer Go. Artificial Intelligence 175(11), 1856–1875 (2011)
13. Haussler, D.: Learning conjunctive concepts in structural domains. Machine Learning 4(1), 7–40 (1994)
14. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing 70(1), 489–501 (2006)
15. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Computer Science Department, University of Toronto, Tech. Rep (2009)
16. Kuleshov, V., Precup, D.: Algorithms for the multi-armed bandit problem. Journal of Machine Learning Research 1, 397–422 (2000)
17. Lai, T.L., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in applied mathematics 6(1), 4–22 (1985)
18. LeCun, Y., Cortes, C.: Mnist handwritten digit database. AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist (2010)
19. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 661–670. ACM (2010)
20. Ou, G., Murphey, Y.L.: Multi-class pattern classification using neural networks. Pattern Recognition 40(1), 4–18 (2007)
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research 12, 2825–2830 (2011)
22. Podolak, I.T., Roman, A.: Theoretical foundations and experimental results for a hierarchical classifier with overlapping clusters. Computational Intelligence 29(2), 357–388 (2013)
23. Robbins, H.: Some aspects of the sequential design of experiments. In: Lai, T., Siegmund, D. (eds.) Herbert Robbins Selected Papers, pp. 169–177. Springer New York (1985)

24. Settles, B.: Active Learning, vol. 6. Morgan & Claypool Publishers (2012)
25. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proceedings of the fifth annual workshop on Computational learning theory. pp. 287–294. COLT '92, ACM (1992)
26. Sindhwani, V., Melville, P., Lawrence, R.D.: Uncertainty sampling and transductive experimental design for active dual supervision. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 953–960. ICML '09, ACM (2009)
27. Xu, Z., Akella, R., Zhang, Y.: Incorporating diversity and density in active learning for relevance feedback. In: Proceedings of the 29th European conference on IR research. pp. 246–257. ECIR'07, Springer-Verlag (2007)