Active Learning Literature Survey

Anita Krishnakumar
University of California, Santa Cruz
Department of Computer Science
anita@soe.ucsc.edu

June 05, 2007

Abstract

The most time consuming and expensive task in machine learning is the gathering of labeled data to train the model or to estimate its parameters. In the real-world scenario, the availability of labeled data is scarce and we have limited resources to label the abundantly available unlabeled data. Hence it makes sense to pick only the most informative instances from the unlabeled data and request an expert to provide the label for that instance. Active learning algorithms aim at minimizing the amount of labeled data required to achieve the goal of the machine learning task in hand by strategically selecting the data instance to be labeled by the expert. A lot of research has been conducted in this area over the past two decades leading to great improvements in performance of several existing machine learning algorithms and has also been applied to diverse fields like text classification, information retrieval, computer vision and bioinformatics, to name a few. This survey aims at providing an insight into the research in this area and categorizes the diverse algorithms proposed based on main characteristics. We also provides a desk where different active learning algorithms can be compared by evaluation on benchmark datasets.

1 Introduction

The central goal of machine learning is to develop systems that can learn from experience or data and improve their performance at some task. In many natural learning tasks, this experience or data is gained interactively, by taking actions, making queries, or doing experiments. Most machine learning research, however, treats the learner as a passive recipient of the data to be processed. This passive approach ignores the learner's ability to interact with the environment and gather data. Active learning is the study of how to use this ability effectively. Active learning algorithms have been developed for classification, regression and function optimization and is found to improve the predictive accuracy of several algorithms compared to passive learning.

2 Major Approaches

The three major approaches to active learning algorithms are as follows.

- Pool-based active learning: As introduced in Lewis and Gale (1994), the learner is provided with a pool of independent and identically distributed unlabeled instances. The active learner at each step chooses an unlabeled instance to request the label from the expert by means of a querying function. This is also called as selective sampling.
- Stream-based active learning: The active learner, for example in Freund 1997, is presented with a stream of unlabeled instances, from which the learner picks an instance for labeling by the expert. This can be visualized as online pool-based active learning.
- Active learning with membership queries: Here, as described in Angluin 1988, the active learner asks the expert to classify cases generated by the learning systems. The learner imposes values on the attributes for an instance and observes the response. This gives the learner the flexibility of framing the data instance that will be the most informative to the learner at that moment.

3 Characteristics of Active Learning Algorithms

We have taken into account several key features that have been addressed in many of the proposed active learning algorithms to compare the effect of each characteristic on the overall performance of the algorithm.

3.1 Ranker consideration

Active learning algorithms may or may not depend on a ranker function to pick the training instance for expert labeling. Several algorithms proposed use support vector machines (SVM), logistic regression, naive Bayes, neural networks, etc. In this section we look at active learning algorithms from the perspective of the ranker function used in the data instance selection process.

Lewis and Gale (1994) describe an uncertainty sampling method where the active learner selects instances whose class membership is most unclear to the learner. Different definitions of uncertainty have been used, for example the Query-by-Committee algorithm by Seung et al. (1992), picks those examples for which the selected classifiers disagree, to be labeled by the expert. The authors suggest that their algorithm can be used with any classifier that predicts a class as well as provides a probability estimate of the prediction certainty.

Cohn et al. (1995) describe how optimal data selection techniques can be applied to statistically-based learning algorithms like a mixture of Gaussians and locally weighted regression. The algorithm selects instances that if labeled and added to the training set, minimizes the expected error on future test data. The authors show that the statistical models perform more efficiently and accurately than the feedforward neural networks.

Similar querying functions have been proposed by Tong and Koller (2000), Campbell et al. (2000) and Schohn and Cohn (2000), called Simple which uses SVMs as the induction component. Here, the querying function is based on the classifier. The algorithm tries to pick instances which are the most informative to the SVM - the support vectors of the dividing hyperplane. This can be thought of as uncertainty sampling where the algorithm selects those instances about which it is most uncertain. In the case of SVMs, the classifier is most uncertain about the examples that are lying close to the margin of the dividing hyperplane. Variations of the Simple algorithm - MaxMin and Ratio methods have been proposed by Tong and Koller (2000), which also use SVM as the learner.

Iyengar et al. (2000) present an active learning algorithm that uses adaptive resampling (ALAR) to select instances for expert labeling. In the work described, a probabilistic classifier is used first to determine the degree of uncertainty and then decision trees is used for classification. The experiments considered use the ensemble of classification models generated in each phase or a nearest neighbor (3-nn) as the probabilistic classifier.

Roy and McCallum (2001) describe a method to directly maximize the expected error rate reduction, by estimating the future error rate by a loss function. The loss functions help the learner to select those instances that maximize the confidence of the learner about the unlabeled data. Rather than estimating the expected error on the full distribution, this algorithm estimates it over a sample in the pool. The authors base their class probability estimates and classification on naive Bayes, however SVMs or other models with complex parameter space are also recommended. Baram et al. (2003) provide an implementation of this method on SVMs, and find it to be better than the original naive Bayes algorithm. Logistic regression is used to estimate the class probabilities in the SVM based algorithm.

Baram et al. (2003) propose a simple heuristic based on "farthest-first" travel sequences for active learning called Kernel Farthest-First (KFF). Here the active learner selects that instance which is farthest away from the current labeled set and can be applied with any classifier learning algorithm. The authors present an application of KFF with an SVM.

Mitra et al. (2004) present a probabilistic active learning strategy for support vector machine learning. Identifying and labeling all the true support vectors guarantees low future error. The algorithm uses the k nearest neighbor algorithm to assign a confidence factor c to all instances lying within the boundary, close to the actual support vectors and $1 - c$ to interior points which are far from the support vectors. The instances are then chosen probabilistically based on the confidence factor.

Nguyen and Smeulders (2004) offer a framework to incorporate clustering into active learning. A classifier is developed based on the cluster representatives and a local noise model propagates the classification decision to the other instances. The model assumes that given the cluster label, the class label of data instance can be inferred. The logistic regression discriminative model is used to estimate the class probability and an isotropic Gaussian model is used to estimate the noise distribution, to propagate information of label from the representatives to the remaining data. A coarse-to-fine strategy is used to adjust the balance between the advantage of large clusters and the accuracy of the data representation.

Osugi et al. (2005) propose an active learning algorithm that balances the exploration and exploitation while selecting a new instance for labeling by the expert at each step. The algorithm randomly chooses between exploration and exploitation at each round and receives feedback on the effectiveness of the exploration step, based on the performance of the classifier trained on the explored instance. The active learner updates the probability of exploring in the next phases based on the feedback. The algorithm chooses between the active learners KFF (which explores) and Simple (which exploits), using SVM^{light} as the classifier, with a probability p. If the exploration is a success, resulting in a change in the current hypothesis, then p is maintained with a high value, encouraging more exploration, else p is updated to reduce the probability of exploration.

3.2 Computational complexity

Computational cost is an important factor to be considered in an algorithm. If an algorithm is computationally very expensive, the algorithm might be infeasible for certain real-world applications which are time sensitive. In this section, we consider the time complexities of the above discussed active learning algorithms in finding an optimal instance for labeling.

The active learning algorithms with mixture of Gaussians and locally weighted regression proposed by Cohn et al. (1995) performs more effectively than the feedforward neural networks where computing variance estimate and re-training is computationally very expensive. With the mixture of Gaussians, training depends linearly on the number of data instances, but prediction time is independent. On the other hand, for a memorybased model like locally weighted regression, there is no training time, but prediction costs exist. However, both can be enhanced by optimized parallel implementations.

The authors Tong and Koller (2000) suggest that the Simple margin active learning algorithm is computationally quite efficient. However, improvement gains can be obtained by querying multiple instances at a time as suggested in Lewis and Gale (1994). But the MaxMin and Ratio methods are computationally very expensive.

Active learning with adaptive resampling (Iyengar et al., 2000) is computationally very expensive because of the decision trees used in the classification phase. The number of phases, number of points chosen to be labeled and number of adaptive resampling rounds also add to the computational cost, hence the authors have chosen these parameters based on computational complexity and accuracy considerations.

The computational complexity to implement the algorithm proposed in Roy and McCallum (2001) is described as “hopelessly inefficient”. However, various heuristics approximations and optimizations have been suggested. Some of these approximations are general and some for the specific implementation by the authors on naive Bayes.

The computational complexity of Kernel Farthest-First algorithm of Baram et al. (2003) is quite similar to the Simple margin algorithm. Simple computes the dot product for every unlabeled instance which takes the same time as computing the argmax for KFF.

The probabilistic active support vector learning algorithm proposed by Mitra et al. (2004) is computationally more efficient than even the Simple margin algorithm by Tong and Koller (2000), as presented in the comparison by the authors.

The active learning using pre-clustering algorithm proposed by Nguyen and Smeulders (2004), use the K-medoid algorithm for clustering as it captures data representation better. But the K-medoid algorithm is computationally very expensive when the number of clusters or data points is large. However, some simplifications have been presented to reduce the computational cost.

The algorithm proposed by Osugi et al. (2005), uses the active learners, Simple or KFF. Hence the time complexity depends on those algorithms. Simple and KFF have similar time complexities and hence, this algorithm has the sample complexity of those algorithms per round.

3.3 Density

In real-world applications, the data under consideration might have skewed class distributions. Some classes might have larger number of samples, and hence a higher density than the other classes. Some classes might have very few instances in the dataset and hence a low density. In this section we analyze whether the active learning algorithms in discussion consider the density of the classes in the dataset, while selecting instances for labeling.

The statistical models of Cohn et al. (1995) selects that instance that minimizes the expectation of the learner's variance on future test set. The instance selection method is independent of density considerations.

The Simple algorithm of Tong and Koller (2000), Campbell et al. (2000) and Schohn and Cohn (2000), picks those instances which are close to the dividing hyperplane. Density of the class distribution is ignored here.

The ALAR algorithm of Iyengar et al. (2000) also selects instances for expert labeling ignoring the density of samples, as it only considers the degree of uncertainty of the classifier.

The algorithm of Roy and McCallum (2001) queries for instances that provide maximal reassurance of the current model. Hence, it does not depend on the class density distribution.

The KFF algorithm of Baram et al. (2003) selects those instances that are the farthest from the current set of labeled instances, which does not really take into account the density of samples.

The probabilistic active support vector learning algorithm by Mitra et al. (2004) does not take into account the density while querying for instances.

The data selection criterion of the active learning algorithm with pre-clustering by Nguyen and Smeulders (2004), gives priority to samples which are cluster representatives, and chooses the ones belonging to high density clusters first. Labeling of high density clusters contribute to a substantial move of the classification boundary, and hence the algorithm clusters the data into large clusters initially. And once the classification boundary between the large clusters have been obtained, the parameters are adjusted to obtain finer clustering for a more accurate classification boundary.

The active learning algorithm proposed by Osugi et al. (2005) explores for new instances using KFF, which does not consider the density of the class distribution. The exploitation phase uses the Simple algorithm which again does not consider the density of samples.

3.4 Diversity

Some active learning algorithms can have added advantage if they take into account the diverse nature of instances in the dataset. The classifier developed will perform well when trained with dataset that has different kinds of samples that represent the entire

	Ranker	Computational complexity	Density	Diversity
Algorithm 1	✓	✓	x	x
Algorithm 2	✓	x	x	x
Algorithm 3	✓	✓	x	x
Algorithm 4	✓	✓	x	x
Algorithm 5	x	x	x	✓
Algorithm 6	✓	x	x	✓
Algorithm 7	✓	✓	✓	✓
Algorithm 8	x	x	x	✓

Table 1: Summary of characteristics of the active learning algorithms in study

distribution.

The algorithms by Cohn et al. (1995), Tong and Koller (2000), Campbell et al. (2000) and Schohn and Cohn (2000) do not consider the diversity of the samples in the labeled set used for training the classifier. They just select the examples that optimize their criterion, which is minimizing the variance in Cohn et al.'s model and choosing the most unclear example to the classifier in the Simple algorithm.

The ALAR algorithm by Iyengar et al. (2000) and the algorithm by Roy and McCallum (2001) also do not select instances based on their diversity.

The KFF algorithm by Baram et al. (2003) select those instances that are the farthest from the given set of labeled examples. Intuitively, this picks the instance from the unlabeled which is most dissimilar to the current set of labeled examples used for training the classifier.

The probabilistic algorithm by Mitra et al. (2004) selects samples that are far from the current boundary with a confidence factor c. This kind of helps the active learner to pick instances in the dataset that are diverse in nature.

The active-learning algorithm by Nguyen and Smeulders (2004), selects diverse samples as it gives priority to samples which are cluster representatives, and each cluster represents a different group of data instances.

The algorithm by Osugi et al. (2005), uses KFF in the exploration phase, which considers the diversity of the dataset while selecting the next instance for expert labeling.

3.5 Close to boundary

Instances lying close to the decision boundary generally contribute to the accuracy of the classifier, as in the case of support vector machines. Hence, those samples lying close to the boundary convey a lot of information regarding the underlying class distribution. In this section we see if the algorithms under study consider the instances lying close to the boundary for expert labeling.

The statistical algorithms of Cohn et al. (1995) selects instances that minimize the variance of the learner on the future dataset, and this might be equivalent to picking those instances close to the current decision boundary.

The Simple algorithm described queries for instances that the learner is most uncertain about and this leads to choosing samples that are close to the classifier's decision boundary.

The ALAR algorithm using 3-nn classifier for the first task of the determining the degree of uncertainty, minimizes the cumulative error by choosing the instances that are misclassified by the classifier algorithm in the second task, given the actual labels are those given by the first algorithm. This chooses the samples that are close to the current decision boundary.

The active learning algorithm by Roy and McCallum (2001) tries to pick samples that provide maximum reassurance of the model and hence does not pick examples close to the boundary. The KFF algorithm also does not choose samples close to the boundary, it queries for the sample farthest from current labeled training set.

The algorithm by Mitra et al. (2004) tries to find the support vectors of dividing hyperplane and hence considers the samples lying close to the decision boundary.

The active learning with pre-clustering algorithm, tries to minimize the current error of the classifier. This leads to choosing those samples lying on the current classification boundary as they contribute the largest to the current error.

The algorithm by Osugi et al. (2005) queries for the samples lying close to the boundary during the exploitation phase, using the Simple active learning algorithm.

3.6 Far from boundary

Some active learning algorithms might query for instances that are far from the current decision boundary, as these examples can help to reassure the model as well as give a chance to explore new instances which might be very informative.

The algorithm by Cohn et al. (1995), the Simple algorithm, the ALAR algorithm and the

KFF do not query for samples lying far from the current decision boundary.

The algorithm proposed by Roy and McCallum (2001) queries for examples that reduce the future generalization error probability. This leads to picking examples that reassure the current model and the samples lying far from the boundary are chosen by the algorithm as the learner is most sure about the labels for those samples.

The active learning algorithm by Mitra et al. (2004) queries for samples lying far from the boundary using the confidence factor c, which varies adaptively with iteration.

The algorithm proposed by Nguyen and Smeulders (2004) picks instances that lie close to the boundary for expert labeling, not the ones far away as they do not contribute towards the current error of the classifier.

The algorithm by Osugi et al. (2005) queries for the samples lying far from the boundary during the exploration phase using the KFF algorithm.

3.7 Probabilistic or uncertainty of ranker

Here we consider whether the ranker used in the active learning algorithm is probabilistic and whether the uncertainty of the ranker is used to query for new instances for expert labeling.Most algorithms studied here in this survey use a probabilistic ranker or uncertainty of the ranker to pick the samples for labeling.

The active learning algorithms with statistical models by Cohn et al. (1995) uses probabilistic measures for determining variance estimates. The Simple active learning algorithm use uncertainty sampling to pick the instance that is most unclear to the learner. The ALAR algorithm that uses the ensemble of classifiers generated in the second task is probabilistic and chooses those samples that are misclassified by the learner. In the algorithm by Roy and McCallum (2001) uses probabilistic estimates using logistic regression to calculated the expected log-loss.

The KFF algorithm does not depend on the probabilistic or uncertainty of the ranker.

The algorithm of Mitra et al. (2004) uses the confidence factor c to pick the samples for labeling which is correlated with the selected samples in the labeled training set.

The algorithm of Nguyen and Smeulders (2004) uses logistic regression with a probabilistic framework and also employs soft cluster membership to choose the sample for labeling.

The algorithm by Osugi et al. (2005) considers probability measures for exploring based on feedback.

3.8 Myopic

An algorithm has a myopic approach when it greedily choose for instances that optimize the criterion at that instance (locally), instead of considering a globally-optimal solution. Most active learning algorithms choose a myopic approach as the learner thinks that the instance it selects for the expert to label, is the last instance that the expert is available for labeling.

Most of the algorithms considered in this study adopt a myopic approach as they try to select the instance that optimizes the performance of the current classifier on the future test set. This is a major limitation in case of greedy margin based methods as the algorithm never explores if the examples lying far from the current decision boundary have more information to convey regarding the class distribution, which helps the classifier to become more effective.

The statistical algorithm of Cohn et al. (1995) queries for the instance that minimizes the expected error of the model by minimizing its variance, which is actually myopic in approach. Similarly, the Simple algorithm by Tong and Koller (2000), Campbell et al. (2000) and Schohn and Cohn (2000) query for the instance that current classifier is most unclear about, at each iteration. Roy and McCallum (2001) also choose the instance that reassures the current model. The KFF algorithm by Baram et al. (2003) also chooses the example that is most different from the current dataset.

However, some of these active learning algorithms select multiple instances to request label from the expert at each iteration instead of choosing just one instance. For example, the ALAR algorithm by Iyengar et al. (2000) and the probabilistic algorithm of Mitra et al. (2004) allows the learner to query the expert for labels of multiple samples at each instance. However this does not exactly globally-optimize the problem in hand.

The algorithm of Nguyen and Smeulders (2004) gives priority to the cluster representatives for labeling after an initial clustering. It also prioritizes examples from the high density clusters first for labeling. This gives the algorithm a kind of approach for global optimization by choosing diverse samples and high density cluster samples initially. But in the later stages the proposed method chooses those instances that contribute the largest to the current error.

The algorithm by Osugi et al. (2005) also gives importance to exploring the dataset with a probability *p*, apart from just optimizing the current criterion. A high value of *p* encourages exploration and is maintained with a high value if the current hypothesis changes. Otherwise it is updated to reduce the probability of exploration at the next step. The value of *p* has an upper and lower bound, and hence there is always a chance of exploring and exploiting.

	Close to boundary	Far from boundary	Probabilistic/ uncertainty of ranker	Myopic
Algorithm 1	✓	x	✓	✓
Algorithm 2	✓	x	✓	✓
Algorithm 3	✓	x	✓	not clear
Algorithm 4	x	✓	✓	✓
Algorithm 5	x	x	x	✓
Algorithm 6	✓	✓	✓	not clear
Algorithm 7	✓	x	✓	x
Algorithm 8	✓	✓	✓	x

Table 2: Summary of characteristics of the active learning algorithms in study

Algorithm	Authors	Name
1	Cohn et al. (2000)	Active learning with statistical models
2	Tong and Koller (2000) Campbell et al. (2000) Schohn and Cohn (2000)	Simple Margin Query learning with large margin classifiers Less is More: Active learning with support vector machines
3	Iyengar et al. (2000)	Active learning with adaptive resampling
4	Roy and McCallum (2001)	Active learning with Sampling estimation of error reduction
5	Baram et al. (2003)	Kernel Farthest First
6	Mitra et al. (2004)	Probabilistic active support vector learning algorithm
7	Nguyen and Smeulders (2004)	Active learning with pre-clustering
8	Osugi et al. (2005)	Balancing exploration and exploitation algorithm for active learning

Table 3: Key to Table 1 & 2

4 Conclusion

Active learning enables the application of machine learning methods to problems where it is difficult or expensive to acquire expert labels. Empirical results presented in the studied research papers indicate that active-learning based classifiers perform better than passive ones. In this paper we have presented a survey of several state-of-the-art active learning algorithms as well as the most popular ones. A detailed analysis of each algorithm has been made based on the characteristics of each algorithm, which gives an insight into the features each active learning algorithm considers while querying for instances for expert labeling.

References

[1] ANGLUIN, D. Queries and concept learning. *Mach. Learn. 2*, 4 (1988), 319–342.

[2] BARAM, Y., EL-YANIV, R., AND LUZ, K. Online choice of active learning algorithms, 2003.

[3] CAMPBELL, C., CRISTIANINI, N., AND SMOLA, A. Query learning with large margin classifiers. In *Proc. 17th International Conf. on Machine Learning* (2000), Morgan Kaufmann, San Francisco, CA, pp. 111–118.

[4] COHN, D. A., GHAHRAMANI, Z., AND JORDAN, M. I. Active learning with statistical models. In *Advances in Neural Information Processing Systems* (1995), G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7, The MIT Press, pp. 705–712.

[5] FREUND, Y., SEUNG, H. S., SHAMIR, E., AND TISHBY, N. Selective sampling using the query by committee algorithm. *Machine Learning 28*, 2-3 (1997), 133–168.

[6] IYENGAR, V. S., APTE, C., AND ZHANG, T. Active learning using adaptive resampling. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2000), ACM Press, pp. 91–98.

[7] LEWIS, D. D., AND GALE, W. A. A sequential algorithm for training text classifiers. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1994), Springer-Verlag New York, Inc., pp. 3–12.

[8] MITRA, P., MURTHY, C., AND PAL, S. K. A probabilistic active support vector learning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 3 (2004), 413–418.

[9] NGUYEN, H. T., AND SMEULDERS, A. Active learning using pre-clustering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning* (New York, NY, USA, 2004), ACM Press, p. 79.

[10] OSUGI, T., KUN, D., AND SCOTT, S. Balancing exploration and exploitation: A new algorithm for active machine learning. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 330–337.

[11] ROY, N., AND MCCALLUM, A. Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning* (2001), Morgan Kaufmann, San Francisco, CA, pp. 441–448.

[12] SCHOHN, G., AND COHN, D. Less is more: Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning* (2000), Morgan Kaufmann, San Francisco, CA, pp. 839–846.

[13] SEUNG, H. S., OPPER, M., AND SOMPOLINSKY, H. Query by committee. In *Computational Learning Theory* (1992), pp. 287–294.

[14] TONG, S., AND KOLLER, D. Support vector machine active learning with applications to text classification. In *Proceedings of ICML-00, 17th International Conference on Machine Learning* (Stanford, US, 2000), P. Langley, Ed., Morgan Kaufmann Publishers, San Francisco, US, pp. 999–1006.