Refinement of Observations under Budget Constraints in Active Learning

Nicolas Cebron
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704
Email: ncebron@icsi.berkeley.edu

***Abstract*— This paper addresses a new challenge in active learning: objects are described in different levels of quality and we can obtain a better version of an object at a given cost. This setting is very useful in robotics, where computing and time resources are limited. We aim to improve and label only a few objects under a given budget in an active learning setting.**

I. Introduction

Active learning [3] enables a learner to pose specific queries that are chosen from an unlabeled dataset. These queries are then answered by a noiseless oracle in an iterative fashion and added to the set of labeled training examples. The concept of active learning is very similar to the human form of learning, whereby problem domains are examined in an active manner. Most of the existing work has been focused on finding optimal selection strategies. Recent work deals with the cost of labels; a few works have considered a setting where the oracle provides features instead of labels, where the feature values may have variable acquisition costs [7] [4]. In this paper, we propose a new active learning setting, which is very useful in robotics where we have limited resources to calculate the attributes of an object. We assume that we can describe the object of interest in views of ascending quality. For example, we might have a robot with a low-level camera that records its environment in VGA quality and a high-level camera that has a resolution of several mega-pixels. Calculating all the features on the mega-pixel level is impossible because the computing resources are limited. We aim to handle most of the examples on the low-level and to improve and classify only a few selected examples within an active learning setting.

II. Adaptive Active Learning

A. Active Learning with SVM

Given a set of labeled training data $D = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_m, y_m)$ where $\vec{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$, a linear support vector machine (SVM) is defined in terms of its hyperplane

$$w \cdot \vec{x} + b = 0 \tag{1}$$

and its corresponding decision function

$$f(\vec{x}) = sgn(w \cdot \vec{x} + b) \tag{2}$$

for $w \in \mathbb{R}^N$ and $b \in R$. Among all possible hyperplanes that separate the positive from the negative examples, the unique optimal hyperplane is the one for which the margin is maximized:

$$\max_{w,b} \{ \min_{\vec{x}_i} \{ ||\vec{x} - \vec{x}_i|| : \vec{x} \in \mathbb{R}^N, w \cdot \vec{x} + b = 0 \} \} \tag{3}$$

As the training data is not always separable, a soft margin classifier uses a misclassification cost C that is assigned to each misclassified example. Equation 3 is optimized by introducing Lagrange multipliers α_i and recasting the problem in terms of its Wolfe dual:

$$\begin{aligned} &\text{maximize: } L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j \\ &\text{subject to: } 0 \leq \alpha_i \leq C, \text{and} \sum_i \alpha_i y_i = 0 \end{aligned} \tag{4}$$

All $\vec{x}_i$ for which the corresponding α_i are non-zero are the so-called support vectors. The complexity of the training is $\Omega(m^2)$ in the size of the training set, which motivated several active learning methods to reduce the size of the training set. The most popular approach is uncertainty sampling, which focuses on selecting examples at the classification boundary. In this case, training examples are selected by their proximity to the dividing hyperplane.

$$\min_{\vec{x}} |w \cdot \vec{x} + b| \tag{5}$$

This idea has been established in the works of [5], [2] and [6].

B. Adaption of Observations

We assume that the objects $\vec{x}$ are described in views of ascending quality. For instance, we could compute the features from an image object at different levels of an image pyramid, where G_0 corresponds to the original image. G_0 is then low-pass-filtered and subsampled by a factor of two to obtain the next pyramid level G_1 and so forth. There might of course be other ways to compute a representation of an object at different levels. In this paper, we assume that we have j views of ascending quality $V_u, u = 1, \ldots, j$. The best level is denoted by V_j and the worst by V_1. We describe the object $\vec{x}$ in view j as $V_j(\vec{x})$.

We further assume that all objects are only given in the worst view V_1 at the beginning. Obtaining a better description of the object comes with a cost $c_u, u = 1, \ldots, j$ that is related to the view level. For simplicity, we assume that we have a cost model that is increasing by arithmetic progression. Each training point in the worst view V_1 has a cost of 1 unit[1] and a cost of 2 units on the next level and so forth.

The Adaptive Active Algorithm is described in Algorithm 1. The key idea is to refine those examples that have the largest

Algorithm 1 Adaptive Active Learning

Require: Budget B, Number of iterations n, Number of view levels j

CurSet = Sample 2 examples from each class randomly.
while Current iteration $\leq n$ **do**
    Train SVM with *CurSet*.
    CandidateSet = *CurSet* $\cap$ D.
    Sort examples in *CandidateSet* in ascending absolute distance to hyperplane, according to Equation 5.
    $\{\vec{x}_1, \ldots, \vec{x}_{\lfloor B/n \rfloor}\}$ = Select the first $\lfloor B/n \rfloor$ examples from *CandidateSet*.
    Obtain the labels for $\{\vec{x}_1, \ldots, \vec{x}_{\lfloor B/n \rfloor}\}$.
    Add the upgraded examples $\{V_{u+1}(\vec{x}_1), \ldots, V_{u+1}(\vec{x}_{\lfloor B/n \rfloor})\}$ to *CurSet*.
end while

impact on the classification model. In case of the SVM, those are the examples that are closest to the dividing hyperplane. We obtain the labels and adapt the quality for these examples.

III. Experiments

Each experiment has been been repeated 100 times. In each iteration, we split up the dataset randomly and use 40% for training and 60% for testing. All training instances are first assumed to be unlabeled and have the lowest quality V_1. A batch of examples is selected in each iteration (plotted on the x-axis) and the mean classification error (given the ground truth in the testing data) is plotted on the y-axis. The multiple features dataset from the UCI Machine Learning Repository [1] consists of features of handwritten numerals ('0'-'9').We have computed two views on these objects: the Zernike Moments based on the original image of size 16x16 (Level 2, green line) and of a subsampled image of size 8x8 (Level 1, red line). We compare our Adaptive Active Algorithm (AAA) against random refinement, a strategy that improves a randomly chosen set of examples in each training iteration (dotted line). We have chosen two classification tasks, where it is hard to discriminate the two digits: '2' vs '3' in Figure 1 and '1' vs '7' in Figure 2. As can be seen, the AAA strategy clearly outperforms random improvement of examples. The AAA strategy starts with a cost of 120 (the cost of Level 1) and increases up to cost 200/160 in the last iteration. For the performance at Level 2, we need to invest 240 units from the very beginning. The AAA strategy achieves the best cost/performance trade-off.

[1]This unit could be monetary or time unit.

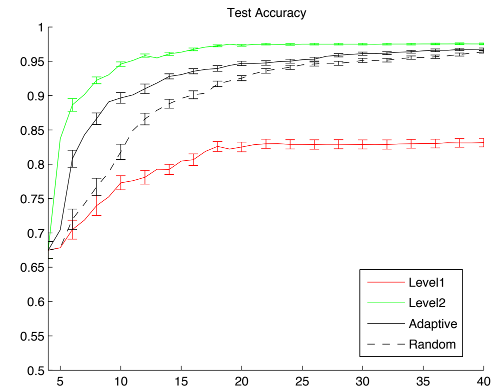

Fig. 1. Test Accuracy '2' vs. '3', Budget 200.

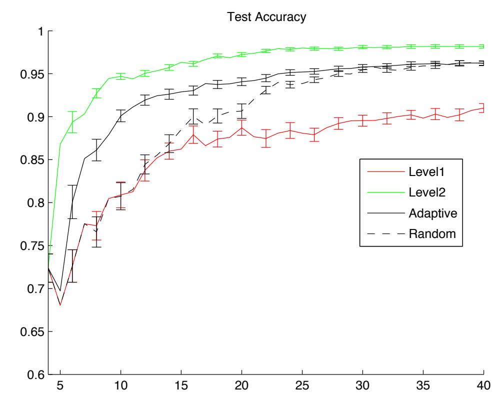

Fig. 2. Test Accuracy '1' vs. '7', Budget 160.

IV. Conclusions

In this work, we have proposed a new active learning setting, where the objects of interest can be obtained at differing quality levels and corresponding costs. We have proposed a new scheme that improves a few selected examples, which clearly outperforms random improvement and provides high classification accuracy with lesser costs.

References

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[2] C. Campbell, N. Cristianini, and A. J. Smola. Query learning with large margin classifiers. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 111–118, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[3] D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[4] M. Saar-Tsechansky, P. Melville, and F. Provost. Active feature-value acquisition. *Manage. Sci.*, 55(4):664–684, 2009.

[5] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In P. Langley, editor, *ICML*, pages 839–846. Morgan Kaufmann, 2000.

[6] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, November 2001. Full version of paper in ICML 2000.

[7] Z. Zheng and B. Padmanabhan. On active learning for data acquisition. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, page 562, Washington, DC, USA, 2002. IEEE Computer Society.