Erschienen in: Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10. - New York, New York, USA : ACM Press, 2010. - S. 1621-1624. - ISBN 978-1-4503-0099-5
https://dx.doi.org/10.1145/1871437.1871688

Active Learning in Parallel Universes

Nicolas Cebron
Multimedia Computing Lab
University of Augsburg
Universitätsstr. 6a, 86159 Augsburg, Germany
cebron@informatik.uni-augsburg.de

Michael R. Berthold
Nycomed Chair for Bioinformatics
University of Konstanz
Box 712, 78467 Konstanz, Germany
michael.berthold@uni-konstanz.de

ABSTRACT

This paper addresses two challenges in combination: learning with a very limited number of labeled training examples (active learning) and learning in the presence of multiple views for each object where the global model to be learned is spread out over some or all of these views (learning in parallel universes). We propose a new active learning approach which selects the best samples to query the label with the goal of improving overall model accuracy and determining which universe contributes most to the local model. The resulting combination and class-specific weighting of universes provides a significantly better classification accuracy than traditional active learning methods.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Theory, Performance

Keywords

Active Learning, Machine Learning, Parallel Universes

1. INTRODUCTION

The goal of inductive machine learning is to learn a model from examples in a dataset that is accurate and generalizes well. In the supervised learning scenario, a set of labeled training examples is used to train a classifier that can be used to predict the target variable for unseen test data. It is common for many real world classification tasks to have a large pool of unlabeled samples available. In many cases the cost of generating a label for an example is high, because it has to be determined by a human expert. Therefore, the expert should be asked to label only a small, carefully chosen subset of the data to train the classifier. Choosing this subset randomly usually requires a large number of samples to improve classification accuracy satisfactorily. Instead of picking random examples, it is preferable to iteratively pick those examples that can help most to improve the classifier's performance. The concept of active learning tackles this problem by enabling a learner to pose specific queries that are chosen from an unlabeled dataset. In this setting, one usually assumes access to a (noiseless) oracle (often a human expert) that is able to return the correct class label of a sample [3].

In the traditional machine learning scenario, the learner has access to the entire set of domain features. However, diverse descriptions for the data objects are often available. Let us consider an example from the domain of object recognition: Typically, we have different feature modules that we can employ to calculate the numerical features (e.g. the shape, histogram or texture) for an image object. Figure 1 shows this situation where an image of a strawberry is described by different feature sets.

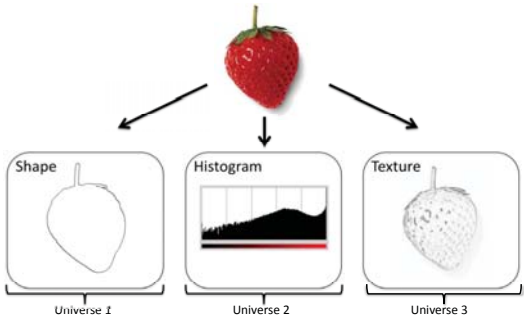

Figure 1: Different sets of features that can be obtained from an image object.

These features are often stringed together to form a long, high-dimensional feature vector. However, such high-dimensional feature vectors cause problems in finding global optima for the parameter space [2], and for wildly diverse types of features this concatenation is a problem in itself. One method of overcoming this problem is feature selection or feature weighting [5]. However, most of these approaches are supervised, relying on a sufficiently large labeled dataset. In many problem settings – such as in our active learning setting – sufficiently labeled data may not be available. In addition, feature selection methods do not make use of the semantics behind having sets of features of different origin. Multi-view learning [8] is one approach to dealing with such different descriptor spaces. However all published approaches assume the existence of one global model, which is deri-

ved in consensus from the models built in each view. In [10] a more flexible learning scheme called *Learning in Parallel Universes* was introduced, which combines local models from one or some of the descriptor spaces to form a global model, applicable to all samples. Now each feature set can be seen as a universe that describes a particular aspect of the objects. In each universe we can learn a specific, local concept and each universe can contribute to a certain degree to the target concept that is to be learned.

The first aim of this paper is to establish the framework of active learning in parallel universes, to derive new and more enhanced selection strategies, and to improve the classification accuracy with few labeled examples.

The second aim in this paper is to measure the quality of a universe with respect to a specific class based on a few labeled examples in an active learning setting. In many real world settings some universes contribute more to a specific class than other universes and some may even be completely irrelevant and should be ignored. This is the main difference to existing multi-view approaches [8], which assume that each view contains the same structural information.

We begin this paper by formalizing the description of an object in parallel universes in Section 2. We will review related work on active learning and multi-view learning in Section 3. In Section 4, we will introduce our new active learning scheme for parallel universes. Experimental evaluation is then carried out in Section 5 before our conclusions in Section 6.

2. TERMINOLOGY AND NOTATION

The numerical data describing each object constitutes a set X of n feature vectors $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ lying in $\mathbb{R}^d$. The training set consists of a large set of unlabeled data points (referred to as samples) $D_U \subseteq X$ and a small set of labeled data points (referred to as examples) D_L, which contains samples from X and their corresponding labels from a set of m possible class labels Y:
$\{< \vec{x}_1, y_1 >, < \vec{x}_2, y_2 >, \ldots, < \vec{x}_n, y_m >\} \subset X \times Y$.

We want to learn a target concept which can be seen as a function $c : X \rightarrow Y$ mapping the instances to the corresponding classes. Based on the labeled examples D_L, a learning algorithm searches for a function $f : X \rightarrow Y$ such that $\forall \vec{x} \in X, f(\vec{x}) = c(\vec{x})$. The set of all possible functions (hypotheses) that are consistent with the labeled examples D_L is called the *Version Space*[6]. In this work, we assume that the classifier function can produce class probabilities in a class vector $\vec{y}_i$ where the j-th entry corresponds to the probability that the sample $\vec{x}_i$ belongs to class y_j.

We extend the notion of the description of a sample $\vec{x}_i$ in a single universe to a description in l different independent universes, $U_1, \cdots, U_l$. $U_k(\vec{x}_i)$ denotes the description of sample $\vec{x}_i$ in universe U_k. We can then rewrite the example as a tuple of samples in each universe with the corresponding classification: $< \vec{x}_i, y_i >=< U_1(\vec{x}_i), \cdots, U_l(\vec{x}_i), y_i >$. For each universe U_k, we now have a classifier $f_k : U_k(X) \rightarrow U_k(Y)$. The final classification decision for a sample $\widehat{f}(\vec{x}_i)$ is usually based on a combination of the classifiers of the different universes. The notion of parallel universes is very general and allows for different classifiers and distance metrics in the respective universes.

3. ACTIVE LEARNING WITH MV

The most related work on active learning with multiple views is the so-called Co-Testing algorithm from [7]. It is depicted in Algorithm 1. It has been slightly modified to match our notation. In each iteration, the algorithm trains

Algorithm 1 Co-Testing Algorithm

Require: Number of iterations n
1: **while** Current iteration $\leq n$ **do**
2: Learn the classifiers $f_1, f_2, \ldots, f_l$ in the universes $U_1, U_2, \ldots, U_l$
3: Let ContentionPoints = $< U_1(\vec{x}_i), \cdots, U_l(\vec{x}_i), ? > \in D_U | \exists i, j f_i(\vec{x}_i) \neq f_j(\vec{x}j)$
4: Let $< U_1(\vec{x}_i), \cdots, U_l(\vec{x}_i), ? >=$ SelectQuery(ContentionPoints)
5: Remove $< U_1(\vec{x}_i), \cdots, U_l(\vec{x}_i), ? >$ from D_U and ask for its label y_j
6: Add $< U_1(\vec{x}_i), \cdots, U_b(\vec{x}_i), y_j >$ to D_L
7: **end while**
8: $\widehat{f}$ =CreateOutputHypothesis$(f_1, f_2, \ldots, f_l)$

a classifier in each universe based on the labeled training data D_L. Based on that information (in this case, the set of samples that are classified differently among the universes), new samples are chosen, labeled, and added to the training data. The final classification decision is based on a combination of the classifiers in the universes.

In [7], three different strategies are presented to select one of the contention points (CP) for labeling:
naive: This strategy chooses at random one of the contention points.
aggressive: This strategy requires that there exists a confidence measure for a classifier $Conf(f_k)$. It chooses as query the contention point $\vec{x}_i$ on which the least confident of the classifiers $f_1, \ldots, f_l$ makes the most confident prediction:

$$\arg\max_{\vec{x}_i \in CP} \min_{k \in \{1,\ldots l\}} Conf(f_k(\vec{x}_i)) \tag{1}$$

This strategy is designed for high accuracy domains, with little or no noise. On such domains, unlabeled examples that are misclassified with high confidence translates into queries that remove significantly more than half of the version space.
conservative: This strategy chooses the contention point on which the confidence of the predictions are as close as possible

$$\arg\min_{\vec{x}_i \in CP}(\max_{g \in \{f_1,\ldots f_l\}} (Conf(g(\vec{x}_i)) - \min_{h \in \{f_1,\ldots f_l\}} Conf(h(\vec{x}_i))). \tag{2}$$

Conservative Co-Testing is appropriate for noisy domains, where the aggressive strategy may end up querying mostly noisy examples.

4. ACTIVE LEARNING IN PU

In the following sections we describe our new active sample selection and parallel universe combination framework. Although we apply the new paradigm of parallel universes to active learning, the general framework follows the multi-view Co-Testing approach from [7].

The motivation behind our sample selection is to take into account the information of all universes, in contrast to the

multi-view approach from [7] where only the most certain and most uncertain view influence the selection criterion.

Entropy is widely used to measure the uncertainty of classifiers and has also been used for sample selection in committee based active learning [4]. Remember that in this setting, we assume that the classifiers can output class probabilities where the class probability for a sample $\vec{x}_i$ for class y_j in universe U_k is denoted by $U_k(\vec{y}_i^j)$. The resulting entropy (denoted as Classifier Uncertainty CU) for a sample $\vec{x}_i$ is calculated as follows:

$$CU(\vec{x}_i) = -\sum_{j=1}^{m}(\sum_{k=1}^{l} U_k(\vec{y}_i^j)) \log_2(\sum_{k=1}^{l} U_k(\vec{y}_i^j)) \quad (3)$$

Intuitively, a very sharply peaked distribution has a very low entropy, whereas a distribution that is spread out has a very high entropy. Therefore, we take the entropy as an uncertainty measurement for a sample. Instead of identifying contention points, we calculate the CU value for all samples and use it as a ranking criterion for sample selection.

If there is a cluster in a region of the data space that causes high classifier uncertainty among the universes, all sample selection schemes are prone to select samples in this region before exploring other samples in the data space that may also be worth considering. We propose to add a term to the ranking criteria for sample selection that takes into account how many labeled examples are located in the neighborhood of the current sample in each universe. This allows covering of the regions of uncertainty with fewer iterations. Based on a distance measure $dist_k$ for Universe U_k, we denote by $\{\vec{x}_a|\vec{x}_a \in D_L\}$ the p nearest neighbors of a sample $\vec{x}_i$ that are in the set of labeled examples D_L. The sample diversity SD is calculated as:

$$SD(\vec{x}_i) = \sum_{k=1}^{l}\sum_{a=1}^{p} dist_k(U_k(\vec{x}_i), U_k(\vec{x}_a)) \quad (4)$$

If a sample is far away from other labeled examples in D_L it will have a higher SD value. We normalize both the measure of Classifier Uncertainty CU and SD to the interval of $[0, 1]$. Each sample from the unlabeled dataset D_U is ranked based on the sum[1] of CU and SD. In each iteration, the samples with the highest rankings are chosen for labeling.

Current multi-view approaches allow a global weighting that is based on the confidence of the classifier in each view. To output the final classification decision, each classifier is weighted with its confidence. Our parallel universe approach goes one step further by introducing a confidence measure for each class in each universe. We use a leave-one-out estimator on the current labeled dataset D_L to derive the confusion matrix for all classes in each universe. We refer to the confusion matrix as C where $C_{i,j}$ is the i-th row in the j-th column of the confusion matrix. The confusion matrix of universe k is $U_k(C)$. The entries on the main diagonal of the confusion matrix $C_{i,i}$ are the correctly classified examples. For each class j, we calculate the accuracy estimate in universe U_k as the number of correctly classified examples divided by the total number of examples and store the results in the Universe Class Quality (UCQ) matrix:

$$UCQ(k, j) = \frac{U_k(C_{j,j})}{|D_L|} + \frac{1}{l} \quad (5)$$

[1]A weighted linear combination may be considered reasonable, but we did not measure a significant difference.

The second term is a Laplacian smoothing term with the number of universes l to take into account the classes that have not been formed in the current universe, especially during the first iterations. We want to make sure that each universe has the same influence on the final classification decision. Therefore, we normalize the entries of the rows of UCQ to make sure that the sum of class weights sums up to 1:

$$UCQ(k, j) = UCQ(k, j) \cdot \frac{1}{\sum_{j=1}^{m} UCQ(k, j)} \quad (6)$$

The classifiers in each universe need to be combined to derive a global classification for a new sample $\vec{x}_i$. We let each classifier vote on the class probability, weighted by the corresponding Universe Class Quality:

$$\widehat{f}(\vec{x}_i) = \arg\max_{y_j} U_k(\vec{y}_i^j) \cdot UCQ(k, j) \quad (7)$$

The classification incorporates the class probability for a sample in each universe as well as the universe class quality and therefore favors confident classification decisions in high quality universes.

5. 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. After initialization with two randomly selected examples from each class, each active learning scheme selects a batch of five examples in each iteration (plotted on the x-axis) and we look at the mean classification error (given the ground truth in the testing data). We also plot the standard error for each method in each iteration. As a base classification method, we used the K-nearest neighbor (KNN) with $K = 3$ neighbors. We compare our method (*PU:Entropy*) against the three selection schemes (*MV:Random*,*MV:Aggressive*,*MV:Conservative*) that we have introduced in Section 3. We also use entropy to estimate the confidence of the classification in each view $Conf(f_k)$ for this approach. The lower baseline is a complete random selection (*Random*) of samples; the upper baseline is the classification error based on the complete training set with universe class weights (*All Examples*). We also report the error without universe combination for a classifier that is based on the complete training set and all attributes.

We have created a webpage[2] with more experiments on different datasets, further details and the code that have been used in this work.

5.1 Multiple Features Dataset

The multiple features dataset from the UCI Machine Learning Repository [1] consists of features of handwritten numerals ('0'-'9') extracted from a collection of Dutch utility maps. Two hundred patterns per class (for a total of 2,000 patterns) have been digitized in binary images. These digits are represented in terms of the following six feature sets (universes): Fourier coefficients of the character shapes, Profile Correlations, Karhunen-Love coefficients, Pixel Averages in 2 x 3 windows, Zernike moments, and Morphological Features (*mor*). The feature sets are described in more detail in [9].

[2]http://icsi.berkeley.edu/~ncebron/pulearning

The test errors of the different methods are shown in Figure 2. In [9], several results are reported for different com-

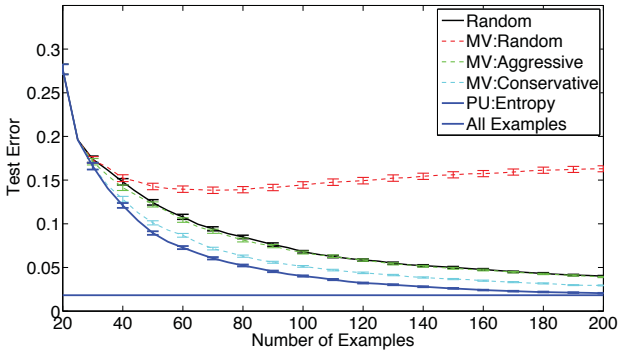

Figure 2: Test Error Multiple Features Dataset.

binations of feature sets, classifiers, and classifier combination methods. They also joined the morphological features and the Zernike moments in one feature set. The best mean results vary from 1.7% to 2.4%. We have used all feature sets and the K-nearest neighbor classifier. The test error of a KNN classifier based on the whole training set is 2.64%; the test error of our parallel universe classifier based on the whole training set is 1.83%. This shows that the class-specific weighting of the universes improves the performance.

The *MV:Random* strategy performs worst with even decreasing performance in later iterations. The *MV:Aggressive* and *MV:Conservative* strategies manage to decrease the test error during the learning iterations but only the *MV:Conservative* is better than complete random selection and both perform significantly worse than our *PU:Entropy* scheme. We make the following observations for the multiple features dataset: The Zernike and the Fourier features have a low weight for class '6' and '9', which corresponds with the finding that these features are rotation invariant.

5.2 Breast Cancer Dataset

The Breast Cancer Wisconsin dataset consists of features from a digitized image of a fine needle aspirate of a breast mass which describe the characteristics of the cell nuclei in the image. There are two classes (malignant and benign). To create different representations of a dataset, we employ 8 different kernels and transformed the kernel matrices to distance matrices so that they can be used with the KNN classifier. The test error is shown in Figure 3. The test error of a KNN classifier based on the whole training set is 4.23%; the test error of our parallel universe classifier based on the whole training set is 4.16%. Our *PU:Entropy* strategy outperforms the other strategies; the *MV:Random* strategy performs worse than complete random selection.

6. CONCLUSIONS

In this paper we addressed the problem of classifying a large unlabeled dataset that is described in different universes with the help of a human expert. We introduced a new active learning paradigm in parallel universes, which combines local models in each universe to decide which sample contributes most to a global classification. Classification of the local models is also used to derive a global classification

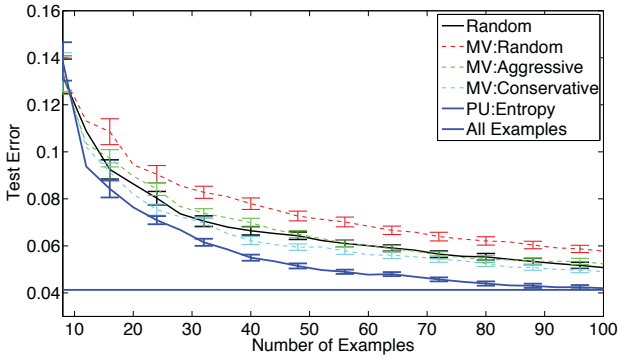

Figure 3: Test Error Wisconsin Breast Cancer Dataset.

decision. In contrast to current approaches we also tracked the quality of a universe with respect to a class with very few labeled examples and integrated this quality measure in the selection and classification of samples. Experiments have shown that this helps to improve the classification accuracy of an active learning scheme in a setting where several different descriptions of the data are available.

Acknowledgements

This work was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service.

7. REFERENCES

[1] A. Frank and A. Asuncion. UCI machine learning repository, 2010. *http://archive.ics.uci.edu/ml*.

[2] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

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

[4] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[5] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[6] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[7] I. Muslea, S. Minton, and C. A. Knoblock. Active learning with multiple views. *J. Artif. Intell. Res. (JAIR)*, 27:203–233, 2006.

[8] S. Rueping and T. Scheffer, editors. *Proceedings of the ICML 2005 Workshop on Learning with Multiple Views*, 2005.

[9] M. van Breukelen, R. P. W. Duin, D. M. J. Tax, and J. E. den Hartog. Combining classifiers for the recognition of handwritten digits. *1st IAPR TC1 Workshop on Statistical Techniques in Pattern Recognition*, pages 13–18, 1997.

[10] B. Wiswedel, F. Höppner, and M. R. Berthold. Learning in parallel universes. *Data Mining and Knowledge Discovery*, 21(1):130–152, July 2010.