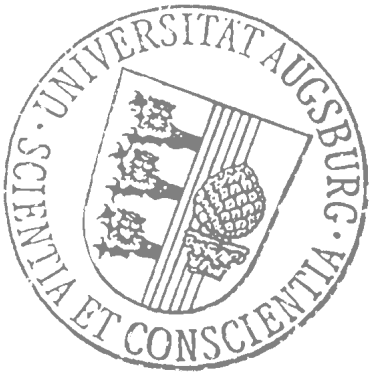
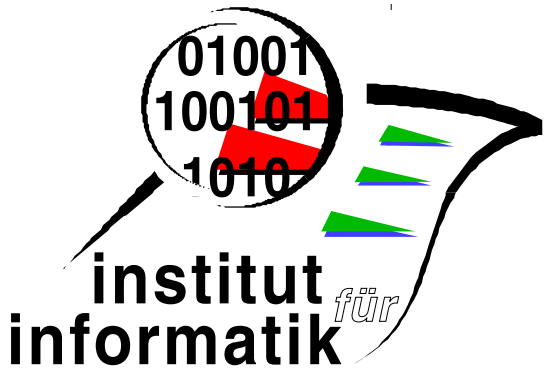
Universität Augsburg



Active Learning in Parallel Universes

Nicolas Cebron and Michael R. Berthold

Report 2011-02 Februar 2011



Institut für Informatik
D-86135 Augsburg

Active Learning in Parallel Universes

Nicolas Cebron and Michael R. Berthold

February 25, 2011

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.

1 Introduction

The goal of inductive machine learning is to learn a model from examples in a dataset that generalizes well and is accurate. 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[1]. 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 [9, 18, 12, 29, 27]. The concept of active learning is very similar to the human form of learning, whereby problem domains are examined in an active manner.

In the traditional machine learning scenario, the learner has access to the entire set of domain features. However, diverse descriptions for the data objects

[1] In this paper we do not address the other aspect of active learning where the learner can actually construct an artificial example to query the label for.

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 for an image object. Figure 1 shows this situation where an image of a strawberry is described by different feature sets. For example, the Zernike features [36] (left) can be used to obtain features that

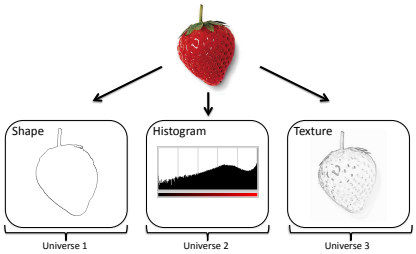


Figure 1: Different sets of features that can be obtained from an image object.

describe the shape of an object. The histogram features (middle) capture the distribution of level intensities for the color red (e.g., the mean value, minimum and maximum value, skewness, kurtosis, and entropy). Texture features (right) – e.g. the Haralick Features [13] – can be used to compute statistics of the co-occurrence of gray level intensities to describe for example, the smoothness or contrast of an image.

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 [3], 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 [16]. 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 (such as the texture, histogram, and shape features in the example above). Multi-view learning [26] is one approach to dealing with such different descriptor spaces. However all published approaches assume the existence of one global model, which is derived in consensus from the models built in each view. In [33] 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. Apart from object recognition a typical example is to classify web pages by either the words on the page or the words contained in anchor texts of links to the page. Also 3D-objects in CAD-catalogs can be described by various feature sets that rely on different

statistics of the object, i.e., projection methods, volumetric representations, 2D images, or topological matchings.

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. Parallel universes can be seen as a committee of classifiers, each model in the universe contributing to the classification concept that is to be learned. The information within each universe can be used to globally select data points that – when being labeled – contribute most to the global classification.

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. Some universes may even be completely irrelevant for specific classes or corrupted by noise and should be ignored. This is the main difference to existing multi-view approaches [26], which assume that each view contains the same structural information. This is an important difference for many real-world settings, most notably the object recognition and molecular data mining domains [32, 33].

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, multi-view learning, and parallel universes 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 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*[19]. 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.

There are many overlaps with respect to existing techniques and terminologies that deal with multiple sets of features. The multi-view learning approach [26] focuses on finding global models in (or across) all views, where it is assumed that in princpiple each view suffices for learning. Our approach differs from existing multi-view learning methods by relaxing the constraint that each view contains the same structural information. The notion of parallel universes [4, 33] allows a part of the model to be created in each universe. During model construction, the different object representations and the intermediate models in each universe aid model construction in other universes and in the end the local models in each universe combined form the global model.

3 State of the Art

3.1 Active Learning

An active learner is described by its underlying classifier and its query function. The classifier is trained on the labeled data. The query function makes a decision based on the current model as to which samples from the unlabeled data pool should be chosen for labeling. We can categorize the existing active learning approaches by their selection strategy:

Optimization of a target function: Based on the minimization of the expected error function (or maximization of a likelihood function) examples can be selected by their contribution to this function. Popular approaches in this field are the works of [18, 25, 10, 15]. From a theoretical point of view the explicit definition of a target function that should be minimized makes it easy to analyze the selection strategy. However, these approaches make several assumptions (e.g., that a stable model built with randomly chosen examples already exists or that the learner does not have a bias [10]); therefore the outcome of the selection strategy depends on how these assumptions apply.

Reduction of version space: The goal of this approach is to reduce the version space with a selected sample as much as possible. One of the most popular approaches is the Query by Committee algorithm [12], which uses a committee of diverse but consistent hypotheses and queries examples for which the disagreement is maximal. Another approach imitates the most general and most specific hypothesis with a neural network and queries examples at the region of uncertainty between those two hypotheses [9]. In the work of [29] the parameter space of a Support Vector Machine (SVM) is related to the version space in order to derive several strategies to query new examples.

Uncertainty sampling: This heuristic approach focuses on selecting examples at the classification boundary. The most popular approaches use an SVM and query examples at the decision hyperplane in the kernel induced space [27, 7] similar to one of the version space reduction approaches described by [29]. Uncertainty sampling is prone to select outliers. Like all other approaches it relies on a stable classification model that has been initialized with some randomly chosen examples.

Several meta-techniques exist to select between different active learning methods to choose an optimal selection strategy in each iteration. In [8], an active learning strategy that balances exploration and exploitation with a prototype based classifier is introduced. In the work of [2] a combination of different active learning algorithms that focus either on the current classification boundary or on exploration have been used together with an algorithm for the multi-armed bandit problem and a novel performance evaluation measure. In [23] a Kernel-Farthest-First algorithm is used for exploration in active learning with SVM. The choice of an exploration step depends on the change that is induced with the newly labeled example on the hypothesis space.

3.2 Multi-View Learning

Multi-view learning methods have been studied in unsupervised [35] and semi-supervised learning settings [6, 21]. In these works it has been noticed that having multiple representations can improve classification performance when, in addition to labeled examples, many unlabeled samples are available. The assumption that all views contain the same structural information is typical for multi-view methods.

The Co-Training algorithm described by [6] uses a small initial training set to learn a classifier in each view. In each iteration, unlabeled samples are classified, and the examples with the highest classification confidence are added to the training set. The individual classifiers of each view are combined with a voting scheme to obtain the final prediction. In the Co-EM algorithm in [21] the samples are classified probabilistically and interchanged between the different views. Both approaches use the knowledge acquired in one view to train the other view.

The work of [20] describes an active learning approach with multiple views. The so-called "contention points" (samples that are classified differently by each view) are queried to improve the classification model. This can be seen as a multi-view uncertainty sampling strategy in active learning. In this work, 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:

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

The authors of [20] state that this strategy is "designed for high accuracy domains, in which there is little or no noise. On such domains, discovering 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

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

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

In this work, "weak views" are introduced, which are only able to learn a concept that is strictly more general or more specific than the target concept.

3.3 Parallel Universes

Learning in parallel universes might be viewed as a variant of multi-view learning because both approaches start with multiple views on the data. The main difference is that the multi-view models currently developed in the literature all induce a single global model that is global in all views/universes, whereas parallel universes combines local models from individual views/universes into a model that is global across all universes. The resulting model is therefore not applicable in each individual universe, but only across all universes. In order to emphasize this difference, the term *Learning in Parallel Universes* was introduced in [4].

Methods for learning in parallel universes have mostly been published in the field of clustering. Extensions of well-known clustering methods like DBSCAN [14], Fuzzy Clustering [24, 33], k-means, k-medoids, and EM [5] have been proposed with promising results. A supervised clustering technique for parallel universes has been proposed in [34]. The focus lies on a model for a particular (minor) class of interest by constructing local neighborhood histograms, so-called Neighborgrams for each object of interest in each universe. Although the algorithm is powerful for the modeling of a minority class, it suffers from computational complexity on larger data sets.

4 Active Learning in Parallel Universes

Although we apply the new paradigm of parallel universes to active learning, the general framework follows the multi-view Co-Testing approach from [20], which will be explained in more detail in the next section. In the following sections we describe our new active sample selection and parallel universe combination framework.

4.1 Base Algorithm

The general Co-Testing algorithm from [20] is depicted in Algorithm 1. It has been slightly modified to match our notation. In each iteration, the algorithm

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$(f_1, f_2, \ldots, f_l)$

trains 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 the next sections we will address how we select new samples for labeling (step 3 and 4), how we measure the quality of a universe with respect to a specific class (based on the given training data D_L in step 6), and how we combine the classifier to output a final classification decision (step 8).

4.2 Sample Selection

The motivation behind our sample selection is to take into account the information of all universes, in contrast to the multi-view approach from [20] that has been introduced in Section 3.2 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 [12]. Re-

member 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:

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

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.

An example is shown in Figure 2. From this Figure, we can see that we

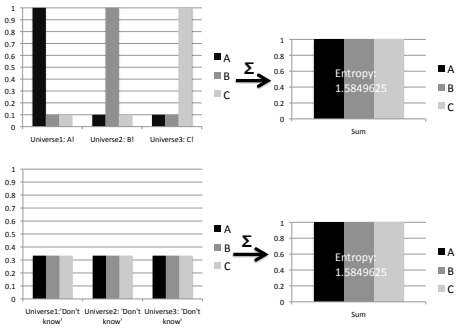


Figure 2: Class probabilities in each universe and the resulting sum of class probabilities.

focus on samples that are either classified differently with high certainty or samples that are classified with high uncertainty among all universes, as both settings result in a high CU value. Instead of identifying contention points, we calculate the CU value for all samples and use it as a ranking criterion for sample selection.

4.3 Relation to Version Space

We define the version space V in parallel universes as the space of all classifiers in their corresponding universe which are consistent with all previously labeled examples.

$$V = \{f_k : \forall \vec{x}_i \in D_L, f_k(\vec{x}_i) = y_i\} \tag{4}$$

As we have seen in the preceding section, our sample selection strategy selects samples in which the resulting sum of class probabilities is very spread out. This can be caused by high contention between the classifiers in the universes and/or high uncertainty in classifying the sample.

Let us look at the first cause: if a sample is selected because there is a high contention between the classifiers, it will split the version space into two parts of comparable size. This is due to the fact that if one of the parts contains most of the version space, then the probability that the classifiers will disagree is very small. Reducing the version space in each active learning iteration is a widely acknowledged idea. However, there is not a guarantee for a fast decrease in test error (especially in noisy problem domains).

If a sample is selected for the second cause - a high uncertainty of the classification decision in each universe - we cannot assume that the version space is split in half. However, these examples have a consolidation effect in either rejecting or strengthening the classifiers.

4.4 Sample Diversity

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:

$$SD(\vec{x}_i) = \sum_{k=1}^{l} \sum_{a=1}^{p} dist_k(U_k(\vec{x}_i), U_k(\vec{x}_a)) \tag{5}$$

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[2] of CU and SD. In each iteration, the samples with the highest rankings are chosen for labeling.

4.5 Universe Class Quality

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. This allows each class to be represented by a combination of universes that is most suitable to the specific class. For example, consider the description of a strawberry in the introduction by its shape, histogram, and texture. If we add two objects, i.e., a tomato and a banana, we may observe that the

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

color information of the histogram is not sufficient to discriminate between the classes strawberry and tomato. However, color histograms can be very useful to separate these two classes from the class banana. Therefore, the histogram universe should be taken into account for the class banana but not for the other two classes.

In an active learning setting, labeled data is hard to come by, so 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. An example for a confusion matrix in one universe is shown in Figure 3. We refer to the confusion matrix as C

Estimated Class / True Class	A	B	C	
A	8	1	1	$\rightarrow\frac{8}{10}$
B	0	9	1	$\rightarrow\frac{9}{10}$
C	3	4	3	$\rightarrow\frac{3}{10}$

Figure 3: Example Confusion Matrix for 3 classes A,B and 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:

$$UCQ(k,j) = \frac{U_k(C_{j,j})}{|D_L|} + \frac{1}{l} \tag{6}$$

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:

$$UCQ(k,j) = UCQ(k,j) \cdot \frac{1}{\sum_{j=1}^{m} UCQ(k,j)} \tag{7}$$

4.6 Universe Combination

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:

$$\widehat{f}(\vec{x}_i) = \arg\max_{y_j} U_k(\vec{y}_i^{\,j}) \cdot UCQ(k,j) \tag{8}$$

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

Before we go into the detailed descriptions of the experiments, we state our experimental methodology. All experiments have been designed such that they are easily reproducible. The algorithms were implemented using the prtools [31] and shogun software [28].

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. The initial training set for all classifiers consists of two randomly selected examples from each class. After initialization, 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. A Bayes estimator on the class frequencies is used to derive the class probabilities.

We compare our method (*PU:Entropy*) against the three selection schemes (*MV:Random*,*MV:Aggressive*,*MV:Conservative*) that we have introduced in Section 3.2 from the multi-view active learning framework described in [20]. 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.

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 (*fou*), Profile Correlations (*fac*), Karhunen-Love coefficients (*kar*), Pixel Averages in 2 x 3 windows (*pix*), Zernike moments (*zer*), and Morphological Features (*mor*). The feature sets are described in more detail in [30].

We have reconstructed the images from the pixel averages dataset. Typical representatives of each class are shown in Figure 4. The test errors of the different methods are shown in Figure 5. In [30], several results are reported

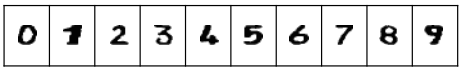


Figure 4: Examples from each class from the Multiple Features Dataset.

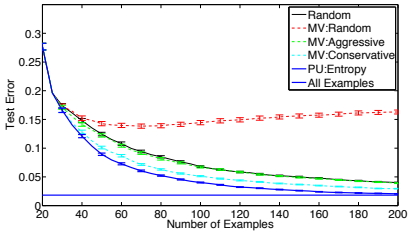


Figure 5: Test Error Multiple Features Dataset.

for different combinations 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.

In Figure 6 we show the first 20 examples that have been selected by our algorithm. Although these examples are different in each iteration - depending

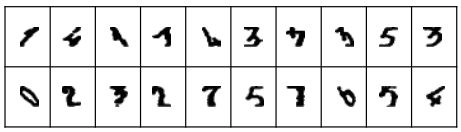


Figure 6: First 20 samples chosen for labeling.

on the random initialization and the data splits - we have observed that the algorithm tends to select examples from classes that are difficult to tell apart, e.g., class '1' and '7'.

We also plot a heat map of the UCQ matrix in Figure 7 to see which universes have been chosen for which class. We make the following observations for the

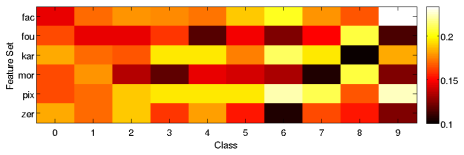


Figure 7: Universe Class Quality for Multiple Features Dataset.

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 Flower Dataset

The flower dataset from [22] consists of images from common flowers in the UK. The images have large scale, pose and light variations and there are also classes with large variations of images within the class and close similarity to other classes. We used the 17 category dataset which contains 80 images per class. Some example images from four different classes are shown in Figure 8, please refer to [22] for a detailed description of the dataset and more examples. The flower dataset consists of four different feature sets that describe the flower

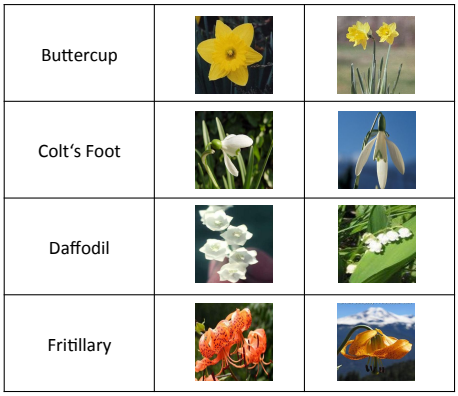


Figure 8: Sample Images from Flowers Dataset.

based on color (HSV), histogram of gradient orientations (HOG) [11], and scale-invariant feature transform (SIFT) [17] sampled on both the foreground region ($SIFTINT$) and its boundary ($SIFTBDY$). The test error of a KNN classifier based on the whole training set is 30.68%; the test error of our parallel universe classifier based on the whole training set is 29.46%. This shows again that the class-specific weighting of the universes is beneficial for the performance.

The test errors of the different active learning methods are shown in Figure 9. In the first iterations, all strategies perform well – the *MV:Conservative* strategy

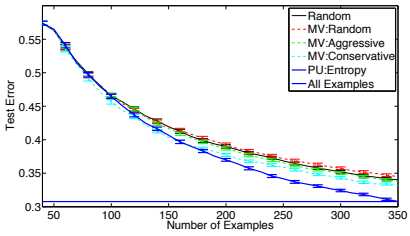


Figure 9: Test Error Flowers Dataset.

slightly better than the others. In later iterations, our *PU:Entropy* consistently outperforms the multi-view selection strategies on the flower dataset.

We also show an example universe class quality matrix for the flowers dataset in Figure 10.

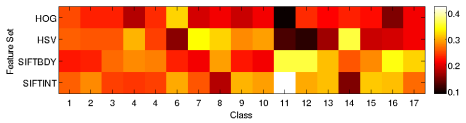


Figure 10: Universe Class Quality for Flowers Dataset.

In Figure 11, we show some examples from class '11' and class '14' as class '11' seems to stand out for its feature sets that describe the shape and texture of the object and class '14' stands out for the color features. From these ex-

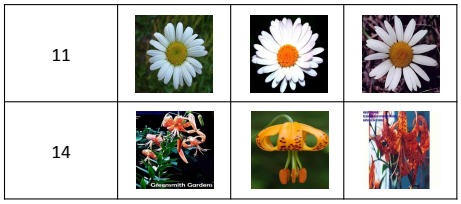


Figure 11: Examples from Class 11 and 14 from Flowers Dataset.

amples, we can comprehend why these different universes have been weighted differently for these classes. Class '11' has a distinct shape and class '14' differs from the other flowers in the color information (most of the other flowers are white/yellow).

5.3 UCI Repository Datasets

Almost all datasets from the UCI repository are represented in a single universe and do not have a meaningful representation in different universes. One could think of partitioning the attributes in different sets. However, there is a very large number of possible partitions and they might not necessarily be semantically meaningful. As the concept of parallel universes is not restricted to attribute sets, we can also compute different representations on a single dataset.

To create these different representations of a dataset, we employ the following kernels: Gaussian, Gaussian Shift, Distance with width 1 and 2, Linear, Sigmoid, and Polynomial kernel with degree 2 and 3. We transformed the resulting 10 kernel matrices to distance matrices so that they can be used with the KNN classifier.

From the UCI Machine Learning Repository [1], we picked three datasets that belong to the category of classification, have numerical attributes and more than 100 instances: the Vehicle, Wisconsin Breast Cancer and Ionosphere dataset.

The Vehicle dataset is based on different descriptors of four different types of vehicle silhouettes (classes). The test error is shown in Figure 12. The

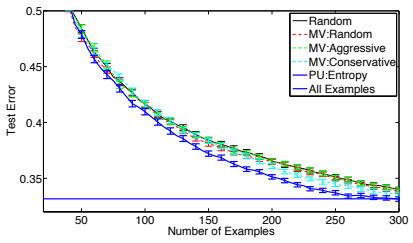


Figure 12: Test Error Vehicle Dataset.

test error of a KNN classifier based on the whole training set is 33.41%; the test error of our parallel universe classifier based on the whole training set is 32.96%. Our *PU:Entropy* strategy performs better than the other strategies. Only the *MV:Conservative* strategy performs better than complete random selection.

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). The test error is shown in Figure 13. 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.

The Ionosphere dataset consists of radar returns from the ionosphere. There are two classes in this dataset: 'good' (radar returns showing evidence of some type of structure) and 'bad' (those that do not). The test error is shown in

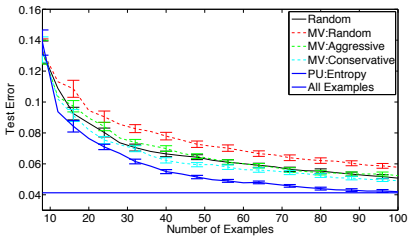


Figure 13: Test Error Wisconsin Breast Cancer Dataset.

Figure 14. The test error of a KNN classifier and of our parallel universe classifier

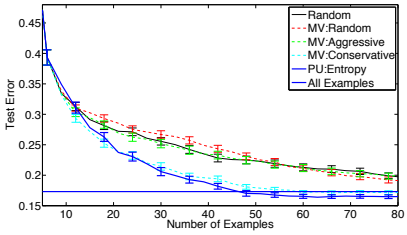


Figure 14: Test Error Ionosphere Dataset.

based on the whole training set is $\approx 18.15\%$. A sparse encoding with prototypes seems to be beneficial, as the test error based on few labeled examples is below the test error on the complete training set. Both our *PU:Entropy* and the *MV:Conservative* strategy perform well on this dataset; the *MV:Conservative* strategy is better in the first iterations whereas our strategy has a better performance in later iterations.

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 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.

References

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

[2] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *J. Mach. Learn. Res.*, 5:255–291, 2004.

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

[4] M. R. Berthold and D. E. Patterson. Towards learning in parallel universes. In *Proceedings of the 2004 IEEE Int'l Conference on Fuzzy Systems (IEEE Int'l Conference on Fuzzy Systems)*, volume 1, pages 67–71. IEEE Press, 2004.

[5] S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, pages 19–26. IEEE Computer Society, 2004.

[6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, New York, NY, USA, 1998. ACM.

[7] 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.

[8] N. Cebron and M. R. Berthold. Active learning for object classification: from exploration to exploitation. *Data Min. Knowl. Discov.*, 18(2):283–299, 2009.

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

[10] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *NIPS*, pages 705–712. MIT Press, 1994.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[12] 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.

[13] R. M. Haralick. Textural features for image classification. *IEEE Trans. System, Man and Cybernetics*, 3(6):610–621, 1973.

[14] K. Kailing, H.-P. Kriegel, A. Pryakhin, and M. Schubert. Clustering multi-represented objects with noise. In H. Dai, R. Srikant, and C. Zhang, editors, *PAKDD*, volume 3056 of *Lecture Notes in Computer Science*, pages 394–403. Springer, 2004.

[15] M. Lindenbaum, S. Markovitch, and D. Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54(2):125–152, 2004.

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

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[18] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Comput.*, 4(4):590–604, 1992.

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

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

[21] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3):103–134, 2000.

[22] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[23] T. Osugi, D. Kun, and S. Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 330–337, Washington, DC, USA, 2005. IEEE Computer Society.

[24] W. Pedrycz. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 23(14):1675–1686, 2002.

[25] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In C. E. Brodley and A. P. Danyluk, editors, *ICML*, pages 441–448. Morgan Kaufmann, 2001.

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

[27] 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.

[28] C. S.Sonnenburg, G.Raetsch and B.Schoelkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

[29] 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.

[30] 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.

[31] F. van der Heijden, R. Duin, D. de Ridder, and a. Tax. *Classification, parameter estimation and state estimation: An engineering approach using Matlab*. Wiley, 2004.

[32] B. Wiswedel and M. R. Berthold. Fuzzy clustering in parallel universes. *Int. J. Approx. Reasoning*, 45(3):439–454, 2007.

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

[34] B. Wiswedel, D. E. Patterson, and M. R. Berthold. Interactive exploration of fuzzy clusters. In J. V. de Oliveira and W. Pedrycz, editors, *Advances in Fuzzy Clustering and its Applications*, pages 123–136. John Wiley and Sons, 2007.

[35] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196, Morristown, NJ, USA, 1995. Association for Computational Linguistics.

[36] F. Zernike. Diffraction theory of the cut procedure and its improved form, the phase contrast method. *Physica*, 1:689–704, 1934.