Empirical Comparison of Active Learning Strategies for Handling Temporal Drift

Mohit Kumar
Flipkart
k.mohit@flipkart.com

Mohak Shah
Robert Bosch LLC
Mohak.Shah@us.bosch.com

Rayid Ghani
University of Chicago
rayid@uchicago.edu

Zubin Abraham
Robert Bosch LLC
Zubin.Abraham@us.bosch.com

ABSTRACT

Active learning strategies often assume that the target concept will remain stationary over time. However, in many real world systems, it is not uncommon for the target concept and distribution properties of the generated data to change over time. This paper presents an empirical study that evaluates the effectiveness of using active learning strategies to train statistical models in the presence of various temporal-drift scenarios. The study also evaluates the benefit of incorporating popular approaches to address temporal drift on the various active learning strategies. The performance of the best performing active learning strategies, were found to be at least comparable, if not significantly better than random sampling strategy across the various types of temporal drifts in 99% of the scenarios tested. In approximately 50% of those instances, active learning strategies were significantly better than random sampling. However, the further away the temporal drift, less is the advantage of using active learning strategies over random sampling. It is shown that uncertainty-based sampling often had the best performance among the various active learning strategies.

1. INTRODUCTION

Active learning algorithms attempt to learn an accurate statistical model by selecting the most informative data to be used for training. The approach is primarily motivated by the fact that in certain domains, labeling of data needed for training a model is expensive. Similar to most other passive learning strategies (where all training examples are labeled), active learning strategies assume that the target concept remains stationary over time [17]. However, many real world data mining applications are deployed in settings that are meant to run for extended periods of time, during which the target concept and data distributions may change. Given that statistical models (such as classifiers that assume the data to be stationary), are known to show reduced accuracy in such *temporal drift* scenarios, there is a need to explore the impact of temporal drift on the active learning strategies used to build these statistical models. Fraud detection, intrusion detection, medical diagnosis, information filtering, and video surveillance are examples of applications that would benefit from this study, given that their labeled examples are expensive to generate and since their domain is prone to temporal drift.

Temporal drift has been categorized into three main types: 'shifting class distribution' (SCD), 'shifting subclass distribution' (SSD) and 'fickle concept drift' (FCD) [7]. SCD is defined to occur when the relative proportion of cases in the different classes may change over time, but the samples within a given class are i.i.d stationary. SSD is defined when a class category may be comprised of a union of (potentially undiscovered) subclasses or themes, and the class distribution of these subclasses may shift over time. FCD refers to the scenario when individual samples may take on different ground truth labels at different times. In this paper, we study two types of drifts, SSD and FCD. We do not study SCD in context of active learning as it is difficult to interpret the results of active learning for SCD, as the difference in performance may be attributed to the underlying change in the class distribution and analyzing the contributions of different active learning strategies may be difficult.

Concept Drift [14] is one form of temporal drift that has been well studied. Concept drift typically refers to the change in the target concept that needs to be learnt over time. There has been work in active learning on streaming data with concept drift [19] and without concept drift [3]. The results from [19] show that random sampling performs better than the proposed active strategies and the authors recommend randomization of active sampling strategies. However, the key difference between streaming data and our focus, is that in the streaming data setup, instances are streaming in to the system and a decision needs to be made right away whether to ask for a label or not. The incoming unlabeled data cannot be stored and queried later. This scenario happens in certain real-world problems (e.g., web search) but is rare in enterprise problems. In most enterprise interactive data mining systems, data needs to be stored anyway for other purposes (e.g., auditing), and the constraint of making a labeling decision instantaneously is not present. Also, in these problems, the domain experts labeling the data are the expensive components of the process and data storage costs often pale in comparison. For these practical reasons, we consider a setting where the unlabeled

KDD 2015 Workshop on Interactive Data Exploration and Analytics (IDEA'15) August 10th, 2015, Sydney, Australia.

pool gets augmented with new data coming in, which is different from the two extreme settings of fixed unlabeled pool and completely stream-based setup with no memory. Chu et al. also mention that a periodically changing unlabeled pool is a more realistic scenario than the two extremes of static unlabeled pool and online streaming data [3]. There are multiple settings for the evolving unlabeled pool. *Cumulative streaming pool* setting is when new unlabeled examples keep coming in and is added to the streaming pool, thus increasing the unlabeled pool available to the learner. *Recent streaming pool* setting is where only the most recent unlabeled examples are available to the learner. In the current work, we only experiment with the *recent streaming pool* setting and leave the *cumulative streaming pool* setting for future work. This corresponds to the *Daily Classification Task* setup recommended by Forman for studying concept drift [7].

In addition to dealing with periodically changing unlabeled pool, it's also not clear whether traditional instance selection strategies (namely uncertainty and density based) still perform well and help adapt the system in the presence of temporal drift. In learning from data streams with concept drift, the popular approach has been to learn classifiers over different time periods and combine them in weighted ensembles [14, 16, 20]. However, the effectiveness of traditional instance selection strategies in the periodically changing unlabeled pool setup is not well understood and hence, explored in this study.

Zliobaite exhaustively reviews learning approaches under temporal drift [21], including learner adaptivity approaches such as adaptive base learners [10], learners with adaptive parametrization [13], adaptive training set formations and fusion rules of the ensembles [18] that are relevant to the current work. Hoens et al. focus on learning with streaming data where there is both concept drift and class imbalance [9]. The authors highlight that this is an under-researched area and applies to many real-world problems. We take special note of this in our work and specifically address problem settings where there is significant class imbalance and show empirical comparison of approaches with different levels of imbalance.

This paper evaluates existing active learning techniques under various temporal drift scenarios to assess if it's worth the additional effort to implement intelligent sample selection strategies over using simple random-sampling, when obtaining labels for training is expensive and the domain is susceptible to temporal/concept drift. The setup of the modeling choices for handling temporal drift consists of three components: 1) the type of model used - ensemble or single, 2) instance or model weighting scheme, within the different types of models/ensembles and 3) the type and amount of concept drift. The setup also helps answer the additional questions - which if any, sample selection techniques is appropriate for a given type of temporal drift? Does the choice of the best performing strategy depend on the evaluation metric chosen? This paper also explores the impact of adapting techniques developed in the temporal drift literature to active learning strategies.

2. FRAMEWORK FOR EMPIRICAL COMPARISON

We present a framework that allows researchers and practitioners to compare the performance of various active learning techniques under temporal drift in a broad range of real-world problems. Specifically, we focus on problem settings where a classifier periodically provides the experts with a ranked list of cases to review and verify. The general setting is analogous to the *Daily Classification Task* introduced by [7] for studying concept drift. Time is discretized into periods (e.g., days) and of all the new data that comes in during that period, a subset of it is labeled based on the active sampling strategy. For example, the number of audited cases of health insurance claims [8], is close to 2% of all new claims that come in a day.

The analysis in this paper is structured based on the following five parameters that can be determined a priori by a domain expert. The type of drift, the amount of drift, the target class distribution, the evaluation metric of interest and the cost of labeled data. Distinct combinations of the five parameters, results in 144 different problem settings on two real-world problems, 'information filtering' and 'intrusion detection'. For these 144 problem settings, a study of the performance of several algorithms combining active learning strategies with temporal drift motivated strategies is carried out. The active learning strategies evaluated include 'certainty sampling', 'uncertainty sampling', 'density-based sampling', and 'sparsity-based sampling'. The various learner adaptation strategies for temporal drift evaluated include 'single model with instance-level weighting', and 'weighted ensemble models'. The three variants of weighting schemes evaluated are uniform, linear, and exponential.

2.1 Domain characterization

2.1.1 *Type and amount of drift.*

We experiment with two types of drift scenarios, Fickle Concept Drift (FCD) and Shifting Subclass Drift(SSD) [7]. FCD is defined where an individual case may have different class labels at different times. For example, in information filtering system the user's preference for relevant news articles may change over time. This kind of drift can be characterized by rate of change in user's preference over time. Thus the amount of drift is parameterized by the probability of switching from one class of interest to another (randomly selected) for the next time period. Even though it can be argued that the user interest may not switch randomly and there may be a semantic pattern to it, we chose to use random switching to be more general and not to introduce an additional bias factor of semantic pattern. We experiment with drift probabilities of 20%, 50% and 100% (labeled as CD0.2, CD0.5 and CD1.0 respectively in figures) as well as the 'no drift' scenario labeled CD0.0.

SSD happens when the positive or negative class comprises of a union of subclasses, and the distribution of these subclasses shift over time. For instance, in network intrusion detection, certain types of intrusions may show up over time as was described in the KDD Cup'1999 dataset [12]. Consequently, while the feature distribution given a particular subclass may be stationary, the feature distribution of the super-class varies over time, because its mixture of subclasses varies. We parameterize the amount of drift by the frequency of the appearance of new subclasses and the disappearance of old ones. We experiment with two drift amounts: drift occurring every 2nd iteration (labeled as Drift=Low), and drift occurring every iteration (labeled as

Drift=High).

2.1.2 Target class distribution.

Most large-scale enterprise data mining problems exhibit class skewness with the level of skewness varying across domains. We experiment with skewness of 10% and 20% for the 'information filtering' task and 1% and 2% for the 'network intrusion detection' task. Although the natural distribution of intrusion cases is very high for the 'KDD Cup network intrusion' dataset, the typical percentage of intrusion cases is expected to be 1 to 2%, which is widely used in studies employing this dataset [5].

2.1.3 Evaluation metric of interest.

Another important characteristic of real-world problems is the performance metric of interest. The choice of evaluation metric is dependent on the domain and the operating range of interest in the domain. We chose the following metrics to cover a broad range of domains: Area Under ROC (AUC) curve, Precision@1st percentile and Precision@10th percentile. The AUC metric is correlated with the ranking accuracy of examples through the entire range [4] and relevant for applications where the performance over all the examples matters. The precision@Kth percentile, is a more focused metric that helps distinguish the performance on the 'top-k' scored cases, making it more relevant for skewed classification problems.

2.1.4 Cost of labeled data.

The number of cases/samples to label is an important design choice, which is also affected by factors such as the budget for labeling. We experiment with representative batch sizes for labeling in a time period with 10 queries and 100 queries, which corresponds to roughly the number of positive examples expected in new unlabeled batches for the two datasets.

2.2 Learning Strategies

We use Support Vector Machine (SVM) as the base classifier and employ various learning strategies as described below.

2.2.1 Active Learning (Sample Selection) choices.

We experiment with four active sampling strategies and compare it to the baseline strategy of random sampling. These include the classical settings of uncertainty and density based sampling and variants of those settings that have been found useful in skewed classification settings [6]. The variant of uncertainty sampling is certainty sampling, where we sample the examples that the classifier is confident on. For linear SVM classifiers, this is basically the distance from the SVM hyperplane as represented by the SVM score. We sample equally from the most confident positive examples, as well as negative examples, to come up with a balanced training dataset. The variant of density sampling include density outlier sampling and sparsity sampling, where we select the examples that are most sparse (least dense). We also evaluate the passive learning setup, where all the data are assumed to be labeled.

2.2.2 Learner adaptation strategies based on historical data.

When building classification models from labeled data spanning more than one period, there are multiple ways to use the historical labeled data for the learner to adapt [21]. We focus on two popular strategies: The first approach builds separate temporal models from each time window by using only the labeled data from that window and then combine those models using ensemble techniques; and second approach builds a *single model* combining all the data (from all time periods) with instance weighting. The *ensemble models* approach has been popularly used to handle concept drift in recent years [14], where a classifier is learnt for each time period and then combined in a (weighted) ensemble. However, a drawback of ensemble based methods is that they need enough labeled data for each time period to build a reasonably good model. In cases where there is not enough labeled data available for each time period, the ensemble based approach may not be ideal. The *single model* approach makes the model more robust (in the presence of limited training data), but less adaptive to temporal drift. One way to compensate for that is weighting the instances differently, based on the time period they belong to.

We experiment with three weighting schemes, for both historical models (in an ensemble's case) and examples (in a single model case): exponential, linear and uniform. Exponential weighting scheme gives exponentially decreasing weight to history, linear weighting scheme gives linearly decreasing weight to the history whereas uniform gives equal weight to history.

3. DATA GENERATION

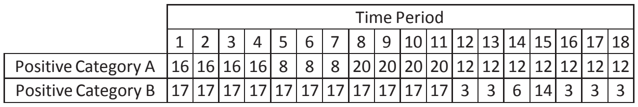
In order to generalize our results and conclusions beyond the data sets we initially used, we use those data sets to then generate variations that span the spectrum in terms of the parameters mentioned earlier. All the datasets along with the relevant parameter details will be made publicly available shortly. For the experiments, we report the results averaged over 10 random splits.

3.1 Fickle Concept Drift

We create FCD datasets based on the '20-Newsgroup' and 'Reuters-RCV1' datasets [15, 2]. For '20-newsgroup', there are 18,774 documents corresponding to 20 news categories after pre-processing and data clean-up. The Reuters-RCV1 dataset is preprocessed as described by [1], where the label hierarchy is reorganized by mapping the data set to the second level of the topic hierarchy. The documents that have labels of the third or fourth level are mapped to their parent category of the second level. The documents that only have labels of the first level are not mapped onto any category. Further, the multi-labeled instances are removed. Out of the resulting 53 second level topics, we select the top 20 most frequent topics and sample 1000 documents for each topic.

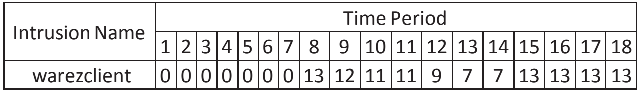
For creating datasets with fickle concept drift, for each time period we sample 50 cases each for the 20 categories for both datasets, resulting in 1000 documents per time period. This gives us 18 time iterations for 20-newsgroup data and 19 time iterations for the RCV1 dataset. We experiment with positive class percentage as 10% (2 out of 20 categories) and 20% (4 out of 20 categories). We test with 0%, 20%, 50% and 100% drift probability over each iteration. Figure 1 shows a sample iteration where the numeric ids correspond to newsgroup categories. For instance, category id '1' corresponds to 'alt.atheism'.

3.2 Shifting Subclass Distribution



	Time Period																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Positive Category A	16	16	16	16	8	8	8	20	20	20	20	12	12	12	12	12	12	12
Positive Category B	17	17	17	17	17	17	17	17	17	17	17	3	3	6	14	3	3	3

Figure 1: A sample iteration of 20-newsgroup data with 10% positive class percentage(2 categories) and 20% drift probability, indicating the positive category id for each time period



Intrusion Name	Time Period																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
warezclient	0	0	0	0	0	0	0	13	12	11	11	9	7	7	13	13	13	13

Figure 2: A sample of the number of cases of subclass 'warezclient' included in 18 time period batches for the KDD cup network intrusion dataset

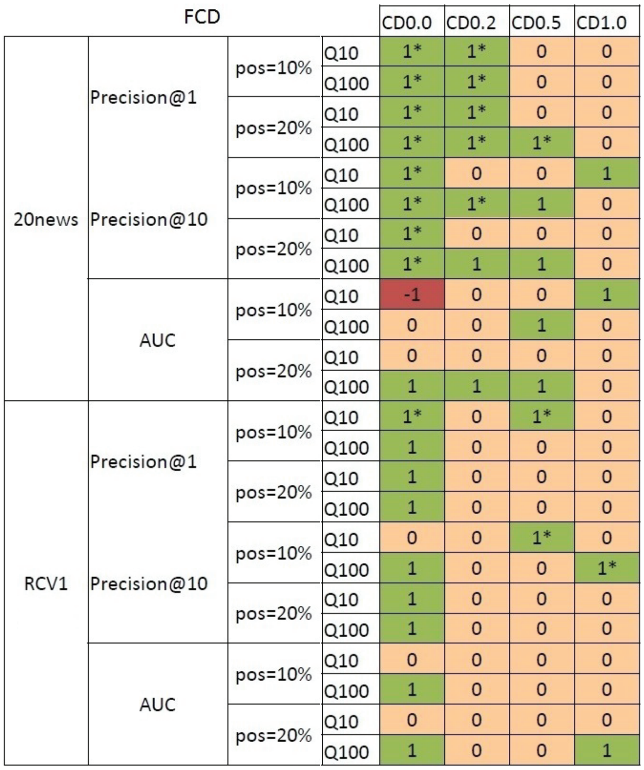
We derive two SSD datasets using '20-Newsgroup' and 'KDD Cup network intrusion detection' [12] datasets. Network intrusion dataset is a natural choice for this type of drift, as different types of intrusions occur at different times. We ignore the training/test split as suggested by the original dataset and instead resample the data according to our strategy to create the temporal dataset.

For each time period, only a subset of the positive classes are present in a batch. We design the sampling scheme such that the first time iteration has only a few subclasses and then new subclasses get added over time, while some existing ones are removed. The exact sampling schemes are not included in the paper in the interest of space; however the sampling scheme will be made publicly available.

Figure 2 shows the number of 'warezclient' intrusion cases included in the batches across 18 time periods for the network intrusion data for *high drift*. There are 40 intrusion subclasses in the dataset, however, we use the 25 most frequent ones. The negative class is predetermined for the network intrusion dataset (subclass: 'normal') and rest of the 24 subclasses are positive (intrusions). We create batches of 8000 datapoints for each time period with positive class varied between 1% and 2%. Sampling for 20-newsgroup is similar, where we have 1000 cases in each time period with positive class varied between 10% and 20%. We arbitrarily select the 'talk' newsgroup category as the positive class with talk.politics.guns, talk.politics.mideast, talk.politics.misc and talk.religion.misc subclasses.

4. RESULTS

The results for the performance metric(s) (AUC, Precision@10 and Precision@1) are computed at each time period, and the average is calculated over all time periods analogous to that of [7]. This performance is averaged over 10 randomized trials with different data samples to come up with a summary evaluation of each learning strategy choice. Namely, the choice of active sampling strategy, the type of model (single vs ensemble), the weighting scheme (for historical instances or models) for various drifts (FCD/SSD with varying amount of drift) and the domain scenarios (class skewness, cost of labeled data). Thus we get a ranking of 30 learning choices for 144 drift/data/performance metric scenarios. While we could choose to evaluate and report results on other measures or measure with finer granularity over Precision@k, the intent in this paper is to cover a reasonable representative range of parameters to observe different trends.



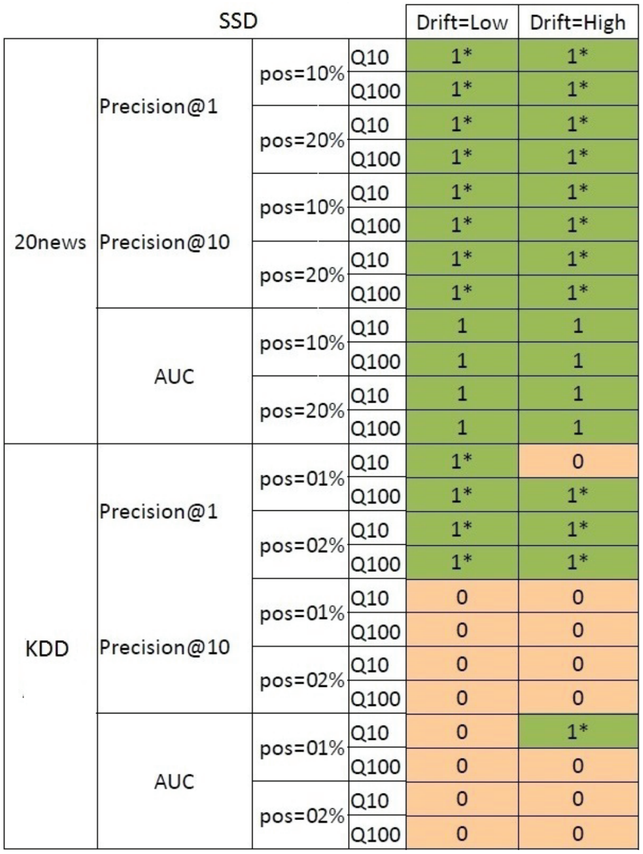
FCD				CD0.0	CD0.2	CD0.5	CD1.0
20news	Precision@1	pos=10%	Q10	1*	1*	0	0
			Q100	1*	1*	0	0
		pos=20%	Q10	1*	1*	0	0
			Q100	1*	1*	1*	0
	Precision@10	pos=10%	Q10	1*	0	0	1
			Q100	1*	1*	1	0
		pos=20%	Q10	1*	0	0	0
			Q100	1*	1	1	0
	AUC	pos=10%	Q10	-1	0	0	1
			Q100	0	0	1	0
		pos=20%	Q10	0	0	0	0
			Q100	1	1	1	0
RCV1	Precision@1	pos=10%	Q10	1*	0	1*	0
			Q100	1	0	0	0
		pos=20%	Q10	1	0	0	0
			Q100	1	0	0	0
	Precision@10	pos=10%	Q10	0	0	1*	0
			Q100	1	0	0	1*
		pos=20%	Q10	1	0	0	0
			Q100	1	0	0	0
	AUC	pos=10%	Q10	0	0	0	0
			Q100	1	0	0	0
		pos=20%	Q10	0	0	0	0
			Q100	1	0	0	1

Figure 3: Performance comparison of the best performing active learning strategy and random sampling. 1 (green colored cells) indicate that best active learning strategy is statistically better than the best random strategy; 0 (orange colored cell) indicates that there is no statistical difference between best active learning strategy and best random strategy and -1 (red colored cell) indicates that the best random strategy is statistically better than the best active learning strategy. $*$ indicates that the performance is 10% better for information filtering task.'pos' is the skew percentage of the positive class and 'Q' the number of queries labeled in a time period.

4.1 Intelligent vs. Random Sampling

Earlier research has shown that random sampling can often outperform active learning strategies under temporal drift [19] when restisted to streaming data. However, unlike the previous study that was restricted to streaming data, the experiments in this study consider the more commonly encountered setting where the unlabeled pool gets augmented with new data coming in.

Figures 3 and 4 show the statistical significance determination results for FCD and SSD under such a scenario. Barring one instance, active learning strategy was comparable or significantly better than random sampling strategy across concept drifts types. Active learning strategy being significantly better almost 50% of the time. We compared the performance by first undertaking a two-way ANOVA omnibus test followed by Bonferroni post-hoc test with 0.05 significance level(p) [11] using the function multcompare in Matlab.



SSD				Drift=Low	Drift=High
20news	Precision@1	pos=10%	Q10	1*	1*
			Q100	1*	1*
		pos=20%	Q10	1*	1*
			Q100	1*	1*
	Precision@10	pos=10%	Q10	1*	1*
			Q100	1*	1*
		pos=20%	Q10	1*	1*
			Q100	1*	1*
	AUC	pos=10%	Q10	1	1
			Q100	1	1
		pos=20%	Q10	1	1
			Q100	1	1
KDD	Precision@1	pos=01%	Q10	1*	0
			Q100	1*	1*
		pos=02%	Q10	1*	1*
			Q100	1*	1*
	Precision@10	pos=01%	Q10	0	0
			Q100	0	0
		pos=02%	Q10	0	0
			Q100	0	0
	AUC	pos=01%	Q10	0	1*
			Q100	0	0
		pos=02%	Q10	0	0
			Q100	0	0

Figure 4: Performance comparison of the best performing active learning strategy and random sampling. 1 (green colored cells) indicate that best active learning strategy is statistically better than the best random strategy; 0 (orange colored cell) indicates that there is no statistical difference between best active learning strategy and best random strategy and -1 (red colored cell) indicates that the best random strategy is statistically better than the best active learning strategy. ∗ indicates that the performance is 10% better for information filtering task (20 newsgroup) and 1% for network intrusion detection task (KDD Cup).'pos' is the skew percentage of the positive class and 'Q' the number of queries labeled in a time period.

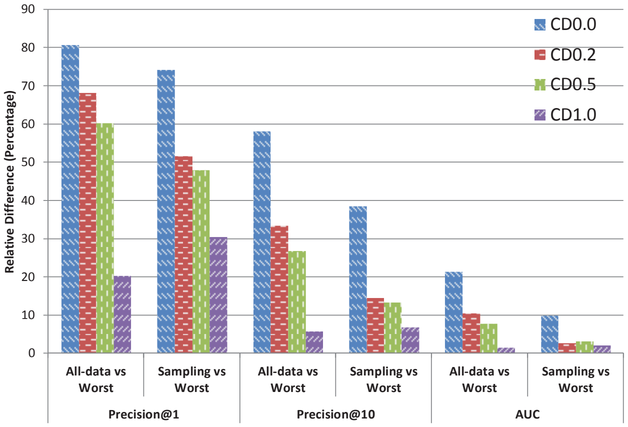
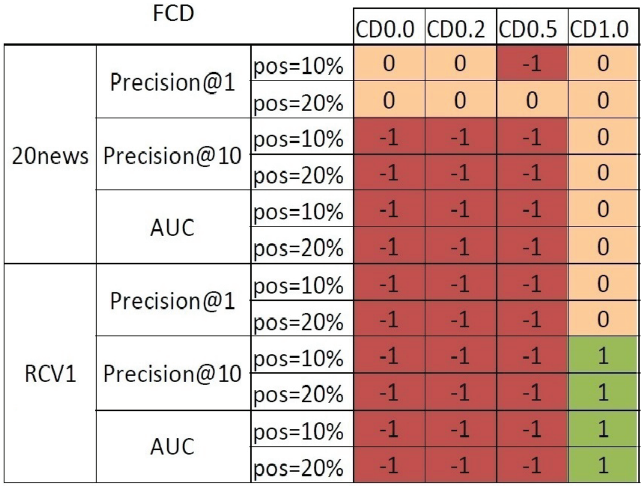


Figure 5: Relative difference between passive learning (using all the data) and best active sampling with the worst performance for the sampling techniques indicating the spread of performance. The positive class percentage is 20% and the number of queries labeled per iteration is 100. Labels CD0.0, CD0.2, CD0.5 and CD1.0 correspond to drift scenarios with probabilities:0, 0.2, 0.5 and 1 respectively

Effect of type and magnitude of drift, on active sampling: Active sampling is the preffered choice over random sampling, in the presence of both high and low magnitude of SSD, as seen in Figure 4. For FCD, which is considered to be a more difficult drift situation [7], the advantage of using active learning strategies wanes as temporal drift increases. The results do however vary across datasets where we observe greater performance difference for the '20-newsgroup' dataset compared to the 'KDD cup' and 'RCV1' dataset. This is intuitively explained by the fact that the subclasses are more closely related to each other in 20-newsgroup (same higher level category 'talk') than the 'KDD cup' dataset, making historical labeled data more useful for '20-newsgroup' than for 'KDD cup'. We conjecture this is because the categories in '20-newsgroup' datasets are more closely related to each other than for RCV1, making historical labeled data more useful for '20-newsgroup' than 'RCV1'. For SSD, we observe that active learning is very useful under different magnitude of drifts for 20-newsgroup whereas only for Precision@1 for KDD cup dataset.

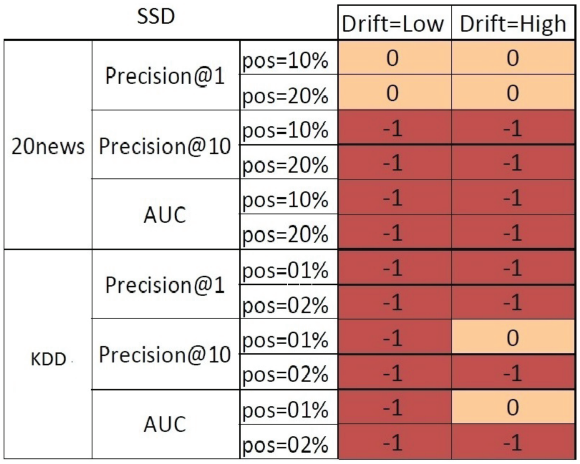
Do the relative performances vary based on the evaluation metric? In general, we observe that performance gain is more pronounced for focused metrics such as Precision@1, whereas for coarser metrics such as AUC, the performance is less variable as observed in Figures 3 and 4. This is observed across various active learning strategies. If the domain of interest has a narrow operating range, such as many real-world problems with class skewness, the difference in performance of active sampling techniques with random sampling is more noticeable.

The relative performance of the worst and best performing sampling strategies gives an indication of the spread of performance and how sensitive the performance is to the choice of sampling strategy. We also compare the best and worst performing sampling choice with the passive learning setup (all the data are labeled and available for training), which both gives us an upper bound on performance and



FCD			CD0.0	CD0.2	CD0.5	CD1.0
20news	Precision@1	pos=10%	0	0	-1	0
		pos=20%	0	0	0	0
	Precision@10	pos=10%	-1	-1	-1	0
		pos=20%	-1	-1	-1	0
	AUC	pos=10%	-1	-1	-1	0
		pos=20%	-1	-1	-1	0
RCV1	Precision@1	pos=10%	-1	-1	-1	0
		pos=20%	-1	-1	-1	0
	Precision@10	pos=10%	-1	-1	-1	1
		pos=20%	-1	-1	-1	1
	AUC	pos=10%	-1	-1	-1	1
		pos=20%	-1	-1	-1	1

Figure 6: Statistical significance comparison between models with 10 actively sampled examples versus 100 randomly selected samples for FCD.



SSD			Drift=Low	Drift=High
20news	Precision@1	pos=10%	0	0
		pos=20%	0	0
	Precision@10	pos=10%	-1	-1
		pos=20%	-1	-1
	AUC	pos=10%	-1	-1
		pos=20%	-1	-1
KDD	Precision@1	pos=01%	-1	-1
		pos=02%	-1	-1
	Precision@10	pos=01%	-1	0
		pos=02%	-1	-1
	AUC	pos=01%	-1	0
		pos=02%	-1	-1

Figure 7: Statistical significance comparison between models with 10 actively sampled examples versus 100 randomly selected samples for SSD.

also gives an indication of the scope of improvement for the different sampling choices and metrics. Figure 5 shows the relative performance difference in percentage between the passive learning (labeled *all-data* in the figure) and the best sampling method (including random) as well as the worst sampling choice for FCD, for the '20-newsgroup' dataset. The major pattern observed is the difference between the best and the worst sampling strategy is large for Precision@1 and reduces progressively for relatively less focussed metrics, such as Precision@10 and AUC.

For all domain settings, the scope for improvement using any intelligent sampling strategy is smallest for Precision@1 and increases for Precision@10 and AUC. However an interesting observation is that when the drift amount is highest (CD1.0) i.e., when in each iteration the positive class is completely changed, the performance of best sampling strategy is better than using *all-data* (comparing the CD1.0 observation across 'All-data vs Worst' and 'Sampling vs Worst' columns). This is probably because for *all-data*, the history is not quite relevant in learning the new class definition. This shows that the history is not useful when drift is extremely high and it is better to use samples of newly obtained data and minimize the use of historical data in learning.

Are the patterns different for different class skewness? There is no significant pattern observable with the different class skewness for comparable data setups for FCD or SSD from Figures 3 and 4.

Are the patterns different for different number of queries per iteration?: There is no significant pattern observable other than that occasionally, the improvement of using active sampling over random sampling was more pronounced with more queries (100) when compared to less number of queries (10) for FCD.

4.2 Practical Considerations

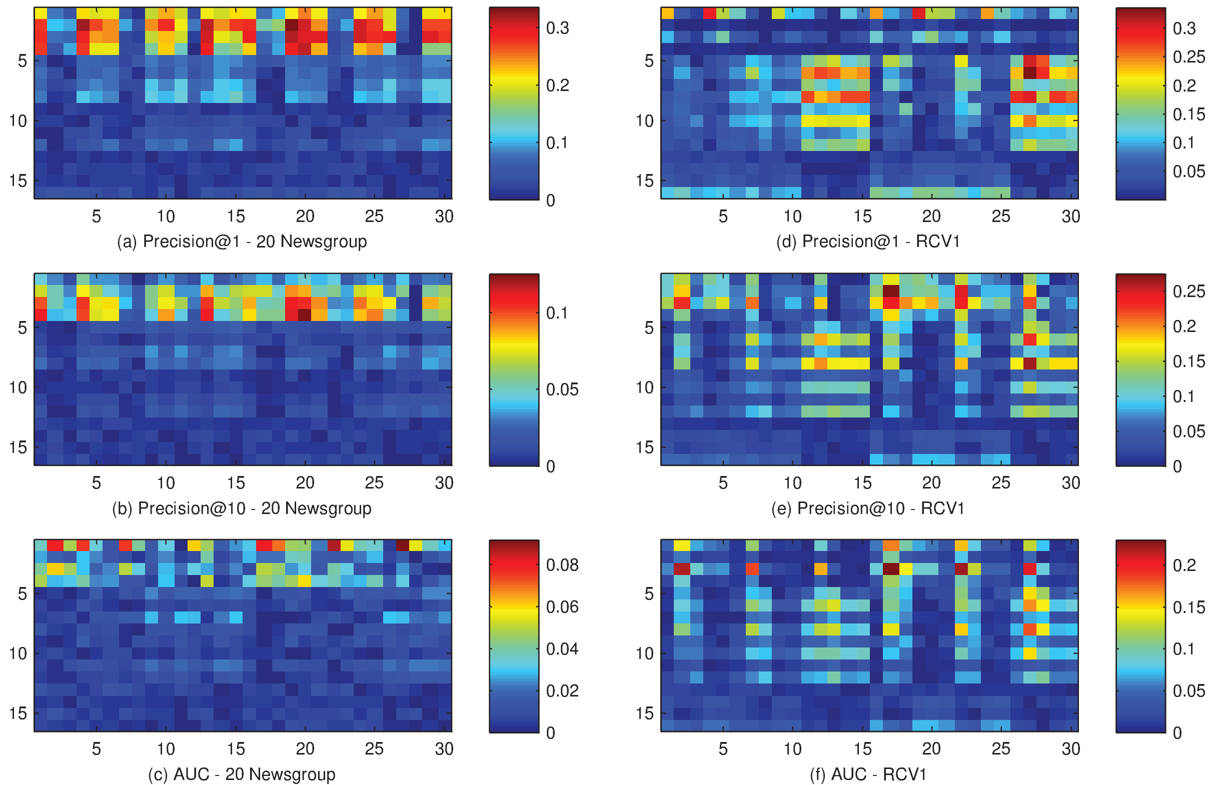
For practical implementation, the actual performance gain achieved is very important in order to justify the value (and added cost of system complexity) of doing active learning. Note that this is not necessarily the same as obtaining a significant difference in a statistical sense. The threshold for the justification of effort may vary across applications. For example, increasing the Precision@1 by 5% can be very significant for applications such as credit card fraud detection, while possibly not be as valuable for information filtering. We choose a threshold of 10% for the 'information filtering' tasks (FCD: 20-newsgroup, RCV1 and SSD: 20-newsgroup) whereas 1% for 'network intrusion' task (SSD: KDD cup) and highlight the results with a '*' in Figures 3 and 4. A cell has a '*', if the difference between the best active sampling strategy and the best random strategy is more than the mentioned threshold.

Is the performance difference significant and worth the cost of implementation? The more focused the evaluation metric, the more significant is the performance difference observed (Figures 3 and 4).

Choosing between labeling more examples randomly or using less labeled examples that are chosen intelligently. We compare the scenario where we label 10 queries using active sampling to the scenario where an order of magnitude more (100 queries) are randomly labeled. Figures 6 and 7 show the results of the statistical significance comparison for both FCD and SSD. We find that labeling more data randomly, almost always gives better performance than intelligent sampling, if the number of samples is one order of magnitude different. One practical implication of this observation is that if the cost of setting up intelligent sampling is high, it may be worth spending the same resources on labeling more randomly sampled data instead. This may in general be true for non-drift situations as well, and may be correlated with sample complexity measures [17], typically used to estimate sample complexity to reach passive learning performance.

4.3 Detailed Results

Figures 8 and 9 show the heatmap of the difference in performance for each active learning strategy relative to the respective best performing choice. The heatmap has separate images for the three different metrics, as the scale



(a) Precision@1 - 20 Newsgroup

(d) Precision@1 - RCV1

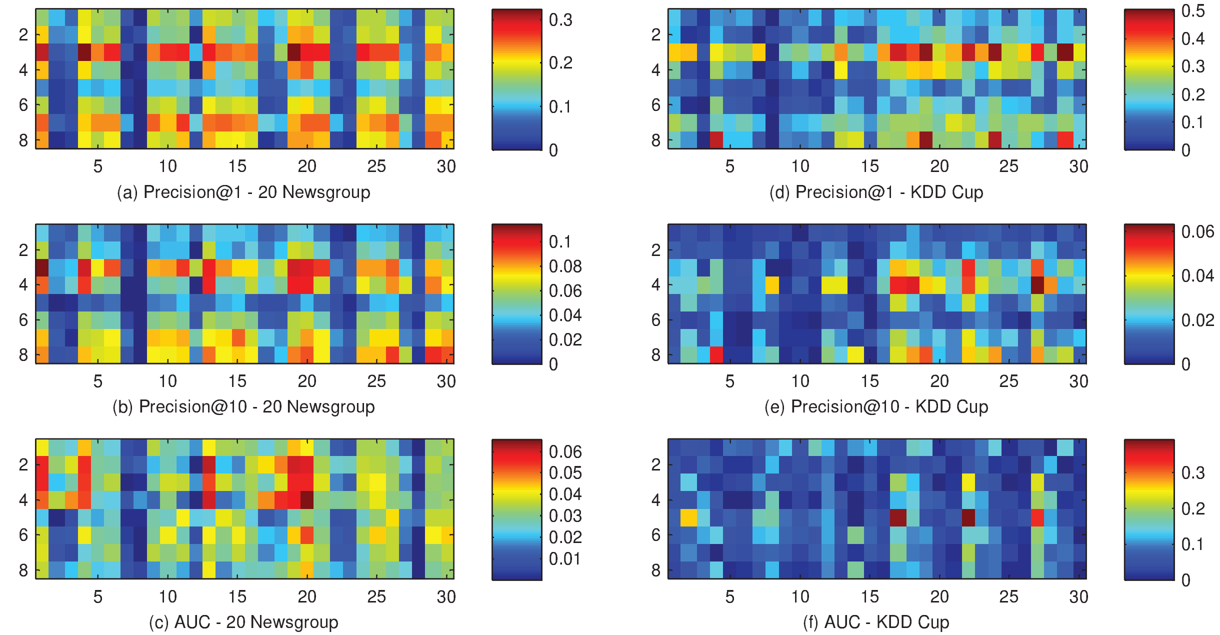
(b) Precision@10 - 20 Newsgroup

(e) Precision@10 - RCV1

(c) AUC - 20 Newsgroup

(f) AUC - RCV1

Figure 8: Heatmap showing the relative performance of the various experimental setups for the different data settings for FCD. Figure 10 shows the respective indexing scheme for the heatmap.



(a) Precision@1 - 20 Newsgroup

(d) Precision@1 - KDD Cup

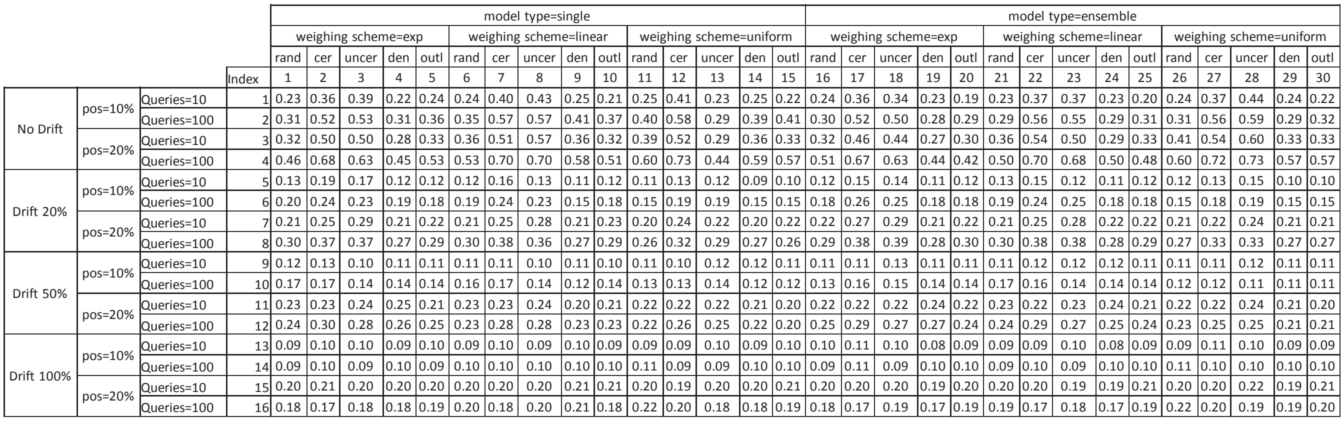
(b) Precision@10 - 20 Newsgroup

(e) Precision@10 - KDD Cup

(c) AUC - 20 Newsgroup

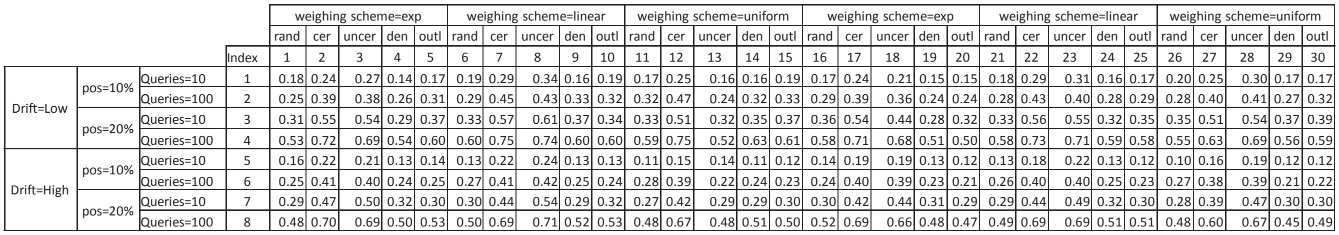
(f) AUC - KDD Cup

Figure 9: Heatmap showing the relative performance of the various experimental setups for the different data settings for SSD. Figure 11 shows the relative indexing scheme for the heatmap.



				model type=single															model type=ensemble														
				weighing scheme=exp					weighing scheme=linear					weighing scheme=uniform					weighing scheme=exp					weighing scheme=linear					weighing scheme=uniform				
				rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl
			Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
No Drift	pos=10%	Queries=10	1	0.23	0.36	0.39	0.22	0.24	0.24	0.40	0.43	0.25	0.21	0.25	0.41	0.23	0.25	0.22	0.24	0.36	0.34	0.23	0.19	0.23	0.37	0.37	0.23	0.20	0.24	0.37	0.44	0.24	0.22
		Queries=100	2	0.31	0.52	0.53	0.31	0.36	0.35	0.57	0.57	0.41	0.37	0.40	0.58	0.29	0.39	0.41	0.30	0.52	0.50	0.28	0.29	0.29	0.56	0.55	0.29	0.31	0.31	0.56	0.59	0.29	0.32
	pos=20%	Queries=10	3	0.32	0.50	0.50	0.28	0.33	0.36	0.51	0.57	0.36	0.32	0.39	0.52	0.29	0.36	0.33	0.32	0.46	0.44	0.27	0.30	0.36	0.54	0.50	0.29	0.33	0.41	0.54	0.60	0.33	0.33
		Queries=100	4	0.46	0.68	0.63	0.45	0.53	0.53	0.70	0.70	0.58	0.51	0.60	0.73	0.44	0.59	0.57	0.51	0.67	0.63	0.44	0.42	0.50	0.70	0.68	0.50	0.48	0.60	0.72	0.73	0.57	0.57
Drift 20%	pos=10%	Queries=10	5	0.13	0.19	0.17	0.12	0.12	0.12	0.16	0.13	0.11	0.12	0.11	0.13	0.12	0.09	0.10	0.12	0.15	0.14	0.11	0.12	0.13	0.15	0.12	0.11	0.12	0.12	0.13	0.15	0.10	0.10
		Queries=100	6	0.20	0.24	0.23	0.19	0.18	0.19	0.24	0.23	0.15	0.18	0.15	0.19	0.19	0.15	0.15	0.18	0.26	0.25	0.18	0.18	0.19	0.24	0.25	0.18	0.18	0.15	0.18	0.19	0.15	0.15
	pos=20%	Queries=10	7	0.21	0.25	0.29	0.21	0.22	0.21	0.25	0.28	0.21	0.23	0.20	0.24	0.22	0.20	0.22	0.22	0.27	0.29	0.21	0.22	0.21	0.25	0.28	0.22	0.22	0.21	0.22	0.24	0.21	0.21
		Queries=100	8	0.30	0.37	0.37	0.27	0.29	0.30	0.38	0.36	0.27	0.29	0.26	0.32	0.29	0.27	0.26	0.29	0.38	0.39	0.28	0.30	0.30	0.38	0.38	0.28	0.29	0.27	0.33	0.33	0.27	0.27
Drift 50%	pos=10%	Queries=10	9	0.12	0.13	0.10	0.11	0.11	0.11	0.11	0.10	0.11	0.10	0.11	0.10	0.12	0.12	0.11	0.11	0.11	0.13	0.11	0.11	0.11	0.12	0.12	0.12	0.11	0.11	0.11	0.12	0.11	0.11
		Queries=100	10	0.17	0.17	0.14	0.14	0.14	0.16	0.17	0.14	0.12	0.14	0.13	0.13	0.14	0.12	0.12	0.13	0.16	0.15	0.14	0.14	0.17	0.16	0.14	0.14	0.14	0.12	0.12	0.11	0.11	0.11
	pos=20%	Queries=10	11	0.23	0.23	0.24	0.25	0.21	0.23	0.23	0.24	0.20	0.21	0.22	0.22	0.22	0.21	0.20	0.22	0.22	0.22	0.24	0.22	0.23	0.22	0.23	0.24	0.21	0.22	0.22	0.24	0.21	0.20
		Queries=100	12	0.24	0.30	0.28	0.26	0.25	0.23	0.28	0.28	0.23	0.23	0.22	0.26	0.25	0.22	0.20	0.25	0.29	0.27	0.27	0.24	0.24	0.29	0.27	0.25	0.24	0.23	0.25	0.25	0.21	0.21
Drift 100%	pos=10%	Queries=10	13	0.09	0.10	0.10	0.09	0.10	0.09	0.10	0.09	0.10	0.09	0.09	0.09	0.10	0.09	0.10	0.10	0.11	0.10	0.08	0.09	0.09	0.09	0.10	0.08	0.09	0.09	0.11	0.10	0.09	0.09
		Queries=100	14	0.09	0.10	0.09	0.10	0.09	0.10	0.10	0.10	0.10	0.10	0.11	0.09	0.09	0.10	0.10	0.09	0.11	0.09	0.10	0.10	0.09	0.10	0.09	0.10	0.10	0.11	0.10	0.10	0.10	0.10
	pos=20%	Queries=10	15	0.20	0.21	0.20	0.20	0.20	0.20	0.20	0.20	0.21	0.21	0.20	0.19	0.20	0.20	0.21	0.20	0.20	0.20	0.19	0.20	0.20	0.20	0.19	0.19	0.21	0.20	0.20	0.22	0.19	0.21
		Queries=100	16	0.18	0.17	0.18	0.18	0.19	0.20	0.18	0.20	0.21	0.18	0.22	0.20	0.18	0.18	0.19	0.18	0.17	0.19	0.17	0.19	0.19	0.17	0.18	0.17	0.19	0.22	0.20	0.19	0.19	0.20

Figure 10: Table indicating the indexing scheme for the Heatmaps in Figure 8. The data shown in the table correspond to Figure 8(a) - Precision@1 for 20 Newsgroup dataset. The abbreviated naming convention for the active learning strategies are: rand - random; cer - certainty; uncer - uncertainty; den - density; outl - density outlier



				weighing scheme=exp					weighing scheme=linear					weighing scheme=uniform					weighing scheme=exp					weighing scheme=linear					weighing scheme=uniform				
				rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl	rand	cer	uncer	den	outl
			Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Drift=Low	pos=10%	Queries=10	1	0.18	0.24	0.27	0.14	0.17	0.19	0.29	0.34	0.16	0.19	0.17	0.25	0.16	0.16	0.19	0.17	0.24	0.21	0.15	0.15	0.18	0.29	0.31	0.16	0.17	0.20	0.25	0.30	0.17	0.17
		Queries=100	2	0.25	0.39	0.38	0.26	0.31	0.29	0.45	0.43	0.33	0.32	0.32	0.47	0.24	0.32	0.33	0.29	0.39	0.36	0.24	0.24	0.28	0.43	0.40	0.28	0.29	0.28	0.40	0.41	0.27	0.32
	pos=20%	Queries=10	3	0.31	0.55	0.54	0.29	0.37	0.33	0.57	0.61	0.37	0.34	0.33	0.51	0.32	0.35	0.37	0.36	0.54	0.44	0.28	0.32	0.33	0.56	0.55	0.32	0.35	0.35	0.51	0.54	0.37	0.39
		Queries=100	4	0.53	0.72	0.69	0.54	0.60	0.60	0.75	0.74	0.60	0.60	0.59	0.75	0.52	0.63	0.61	0.58	0.71	0.68	0.51	0.50	0.58	0.73	0.71	0.59	0.58	0.55	0.63	0.69	0.56	0.59
Drift=High	pos=10%	Queries=10	5	0.16	0.22	0.21	0.13	0.14	0.13	0.22	0.24	0.13	0.13	0.11	0.15	0.14	0.11	0.12	0.14	0.19	0.19	0.13	0.12	0.13	0.18	0.22	0.13	0.12	0.10	0.16	0.19	0.12	0.12
		Queries=100	6	0.25	0.41	0.40	0.24	0.25	0.27	0.41	0.42	0.25	0.24	0.28	0.39	0.22	0.24	0.23	0.24	0.40	0.39	0.23	0.21	0.26	0.40	0.40	0.25	0.23	0.27	0.38	0.39	0.21	0.22
	pos=20%	Queries=10	7	0.29	0.47	0.50	0.32	0.30	0.30	0.44	0.54	0.29	0.32	0.27	0.42	0.29	0.29	0.30	0.30	0.42	0.44	0.31	0.29	0.29	0.44	0.49	0.32	0.30	0.28	0.39	0.47	0.30	0.30
		Queries=100	8	0.48	0.70	0.69	0.50	0.53	0.50	0.69	0.71	0.52	0.53	0.48	0.67	0.48	0.51	0.50	0.52	0.69	0.66	0.48	0.47	0.49	0.69	0.69	0.51	0.51	0.48	0.60	0.67	0.45	0.49

Figure 11: Table indicating the indexing scheme for the Heatmaps in Figure 9. The data shown in the table correspond to Figure 9(a) - Precision@1 for 20 Newsgroup dataset. The abbreviated naming convention for the active learning strategies are: rand - random; cer - certainty; uncer - uncertainty; den - density; outl - density outlier

of differences is very different for the three metrics. '0' value in the heatmap relates to the best performing modeling scheme. The larger the value in this heatmap for the modeling strategy the worse it performs. The data in Figure 10 corresponds to raw values (prior to normalization) used to generate Figure 8(a) and the data in Figure 11 corresponds to Figure9(a). For instance, for Figure 8(a), row index 3 corresponds to row 3 of Figure 10, where the data has 'No Drift'; percentage of positive examples is 20% and number of queries per time period is 10. The best performing learning choice is for index '28', which corresponds to an ensemble model with uniform instance weighting scheme and using 'uncertainty' active sampling strategy.

Which active sampling strategy in general performs better? In general, uncertainty sampling is the best active sampling choice. For SSD, the second best choice for active sampling is certainty based sampling. Density based sampling is not well suited for SSD.

Which type of model in general performs better? For SSD, the best performing models are single models rather than ensembles. For FCD, ensemble models perform better than single models. The intuitive explanation is that since the true concept is not changing for SSD (only the subclass frequency within the broader concept is changing), learning a single model that represents the concept fares better. For FCD, where the true concept is changing, learning models for different time periods which represent the concept for that time period helps by possibly learning disjoint concepts, which is not possible with a single linear model.

Which weighting scheme in general performs better? For FCD, the exponential weighting scheme works better than the linear and uniform weighting schemes for both single and ensemble model types. For SSD, a linear weighting scheme works better than the exponential and uniform weighting schemes. This difference in weighting scheme for FCD and SSD seems intuitive because for SSD, history is more useful than for FCD and forgetting the history slowly (linearly) helps for SSD whereas forgetting the history faster (exponentially) helps for FCD.

These results provide crucial insights indicating that the optimal design choices for interactive systems, need to consider broader domain parameters rather than adopting a 'best practice' strategy. A sampling strategy that focuses on detecting the drift and then explicitly sample examples to reflect it, will perform better than drift-agnostic (or random) strategies.

5. CONCLUSION

We find that active sampling performs statistically better than random sampling in nearly half the temporal drift problem setups, while being comparable in the remaining setups. Further away the drift take the subclasses, less advantageous is using active learning stategies. Performance gain is more prominent for focused metrics and less so for aggregated metrics. In general, uncertainty based sampling strategy was the best active learning strategy. We found that the techniques developed in temporal drift literature namely instance weighting and weighted ensembles combined with active learning gave better results. Some intuitive patterns that were validated by the results were: a) ensemble models are better for FCD whereas building a single model (with instance weighting) is better for SSD b) exponential weighting scheme is better for FCD whereas linear weighting scheme is more effective for SSD. The optimal design choices for interactive systems in the presence of temporal drift, needs to consider the broader domain parameters rather than adopting a 'best practice' strategy.

6. REFERENCES

[1] R. Bekkerman and M. Scholz. Data weaving: scaling up the state-of-the-art in data clustering. In *Proc of the 17th ACM CIKM*, 2008.

[2] C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[3] W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng. Unbiased online active learning in data streams. In *Proc of the 17th ACM SIGKDD*, 2011.

[4] P. Donmez and J. G. Carbonell. Active sampling for rank learning via optimizing the area under the roc curve. In *Proc of the 31th ECIR*, 2009.

[5] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Applications of Data Mining in Computer Security*. Kluwer, 2002.

[6] Z. Ferdowsi, R. Ghani, and M. Kumar. An online strategy for safe active learning. In *ICML Workshop on Combining Learning Strategies to Reduce Label Cost*, 2011.

[7] G. Forman. Tackling concept drift by temporal inductive transfer. In *Proc of the 29th SIGIR*, 2006.

[8] R. Ghani and M. Kumar. Interactive learning for efficiently detecting errors in insurance claims. In *Proc of the 17th ACM SIGKDD*, 2011.

[9] T. R. Hoens, R. Polikar, and N. V. Chawla. Learning from streaming data with concept drift and imbalance: an overview. *Progress in Artificial Intelligence*, 2012.

[10] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proc of the 7th ACM SIGKDD*, 2001.

[11] N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[12] KDD-Cup. Kdd cup, 1999.

[13] R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In *Proc of the 17th ICML*, 2000.

[14] J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *Journal Machine Learning Research*, 2007.

[15] J. Rennie. 20 newsgroups data set, 2007.

[16] M. Scholz and R. Klinkenberg. Boosting classifiers for drifting concepts. *Intelligent Data Analysis*, 2007.

[17] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[18] W. N. Street and Y. Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proc of the 7th ACM SIGKDD*, 2001.

[19] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with evolving streaming data. In *Proc of the ECML PKDD-part III*, 2011.

[20] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc of the 9th ACM SIGKDD*, 2003.

[21] I. Zliobaite. Learning under concept drift: an overview. *CoRR*, abs/1010.4784, 2010.