Tackling Concept Drift by Temporal Inductive Transfer

George Forman
Hewlett-Packard Labs
Palo Alto, CA, USA
ghforman@hpl.hp.com

ABSTRACT

Machine learning is the mainstay for text classification. However, even the most successful techniques are defeated by many real-world applications that have a strong time-varying component. To advance research on this challenging but important problem, we promote a natural, experimental framework—the Daily Classification Task—which can be applied to large time-based datasets, such as Reuters RCV1.

In this paper we dissect *concept drift* into three main subtypes. We demonstrate via a novel visualization that the *recurrent themes* subtype is present in RCV1. This understanding led us to develop a new learning model that transfers induced knowledge through time to benefit future classifier learning tasks. The method avoids two main problems with existing work in inductive transfer: scalability and the risk of negative transfer. In empirical tests, it consistently showed more than 10 points F-measure improvement for each of four Reuters categories tested.

Categories and Subject Descriptors

H.3.3 [**Information Search & Retrieval**]: Information filtering; I.5 [**Pattern Recognition**]: Design methodology, *Classifier design and evaluation*.

General Terms

Algorithms, Performance, Experimentation.

Keywords

Text classification, topic identification, concept drift, time series, machine learning, inductive transfer, support vector machine.

1. INTRODUCTION

Advanced technology for supervised machine learning is making its way into commercial applications. For example, we have used it to categorize millions of HP technical support documents into hundreds of topic categories for improved customer support. But real-world deployment of techniques that have proven successful in the laboratory often meet with challenging practical problems. Machine learning research typically assumes the manually-labeled training cases are random samples—independently and identically distributed (iid)—from a stationary test distribution. In contrast, commercial applications of machine learning often desire to apply trained classifiers to make predictions on a stream of future samples that may vary over time. Unfortunately, the success of machine learning classification pales for real-world, time-varying streams of data.

Despite the difficulty, this is nonetheless an economically important problem to tackle. Although controlled concept drift scenarios have been devised for individual investigations, the lack of a large, real-world benchmark problem to share, innovate from, and compare against has been a detriment to progress in this area. Towards this end, we define and promote a research framework called the Daily Classification Task (DCT) in which to conduct studies and perhaps competitions. Among other datasets, it can be applied to the large Reuters RCV1 corpus, which has 806,791 news articles over 365 days that are classified into many topics, industries and country categories [14]. It is publicly available from NIST [15], unlike many industrial datasets exhibiting similar concept drift.

We subdivide the notion of *concept drift* into three main types, and demonstrate that Reuters exhibits the *recurrent themes* subtype, illustrated via a new visualization technique. Armed with this understanding, we then go on to develop a classification model that is able to leverage training data from many previous days, even if the target concept has drifted substantially. Finally, we present an empirical DCT evaluation that reveals the strong success of this new model. It effects a form of *inductive transfer* across time, and does so in a way that avoids many of the common problems inherent in the existing vein of research on inductive transfer, as described in the related work section.

2. ANALYSIS OF CONCEPT DRIFT

We subdivide the notion of concept drift into three main types:

1. Shifting Class Distribution: the relative proportion of cases in the different categories may change over time, but the samples within a given class are iid stationary. For example, the proportion of Hepatitis A cases may increase with an epidemic, but the symptoms/features of the disease are invariant over time [3]. Even so, this will change the optimal decision threshold for an imperfect classifier [2]. For a robust method to track shifting class distributions with very limited training data, see [5] or [6].
2. Shifting Subclass Distribution: a category may be comprised of a union of (potentially undiscovered) subclasses or themes, and the class distribution of these subclasses may shift over time. As above, the feature distribution given a particular subclass is stationary, but the feature distribution of the super-class will vary over time, because its mixture of subclasses varies.
3. Fickle Concept Drift: individual cases may take on different ground truth labels at different times. This setting is appropriate for recommender systems—the user may

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-369-7/06/0008...$5.00.

initially find some case relevant that is later not relevant, such as an interest in *Boston weather* that wanes after one's trip there. This appears to be the most difficult setting of class drift. If some assumption can be placed on how slowly or suddenly concepts shift, one may have some notion of how prior training labels may still be useful. In the general case, old training data is no better than unlabeled samples.

A variant on each of these types is when the domain has some recurrent or even periodic behavior. For example, in spam classification, there is a *periodic theme* of Christmas-related spam every December. In the remainder of this paper, we will focus on *recurrent themes*: a subtype of type 2 concept drift, in which we cannot expect strict periodic behavior of resurfacing subclasses, and furthermore, we do not know the subclasses that compose the positive class of interest. Moreover, in problems of interest, new subclass themes may crop up that also belong to the class, but have never before been witnessed.

3. FxTime Visual Analysis

We present here a novel visual analysis of the feature space for binary text classification datasets. Its goal is to determine how stable or shifting the most predictive features are over time. It helps one characterize the degree and nature of concept drift in a dataset.

In each time period independently, we determine the most predictive features—in our case, the top 100 word features for predicting a particular Reuters topic. If the concept is completely stable, the top 100 words will be the same each day; otherwise, they will vary. Given the total set of top words over all time, we sort the words by the date they first appeared in the top 100. We allocate each word to a column of pixels in an image, where each pixel in the column represents one day, going downward. We color a pixel black if the word was among the top 100 predictive words on that day; otherwise we color it grayscale according to its predictiveness, gray being predictive, white non-predictive. Figure 1 shows this visualization for three Reuters topic categories: GCAT (government & social issues, 30% of articles), GSPO (sports, 4.4%), and ECAT (economics, 15%). We selected the top 100 words according to Bi-Normal Separation [7], which like Information Gain, is a feature scoring metric used in feature selection. For each time period, we used a sliding window covering the most recent 7 days of data, in order to avoid weekend-related effects, which otherwise make a distracting horizontal grating pattern that visually obscures other patterns.

First, notice that the top predictive words for ECAT are less stable over time. It has 50% more top words (1081) over the course of the year than either of the other two classes. Later we will see that it also exhibits much lower F-measure. All three images show a few words on the far left that remained predictive throughout the year. For GCAT, these are words such as *police, troops, arrested,* and *peace.*

The frontier of black pixels shows when a feature first makes it into the top 100. Notice by the changes in slope that on some days many new top words are generated as a new hot news topic is introduced, and other times several days pass without many new features, indicating the rate of drift. For all three images, observe that most top words lose their predictiveness after a few days (downward), reinforcing the slogan that yesterday's news is like stale bread. In some cases the popularity of some few predictive keywords lasts several weeks, as in the right-hand oval in GCAT, when *laurent, mobutu, seko, sese,* and *kabila* became predictive in May, 1997, as Laurent Kabila led rebel forces to expel the president of Zaire, Mobutu Sese Seko, from the country.

Interestingly, many or most predictive words resurface again later among the top words. For example, in the left-hand oval, the words *hostage(s), tupac, amaru, mrta,* and *fuimori* were introduced on December 17, 1996, when the Tupac Amaru Revolutionary Movement (MRTA) occupied a Japanese embassy. Later we see this group of words resurface at the bottom of the oval when the Alberto Fujimori regime in Peru massacred 15 MRTA commandos on April 22, 1997. This is a striking example, but there are many other predictive words that come back later when there is additional news on their topic.

Examples like these represent *recurrent themes* of type 2 concept drift: recurrent shifting subclass priors, where we do not know the subclasses. This is not fickle concept drift, because articles that fit a category continue to belong to the class, supposing more news on that topic arises later.

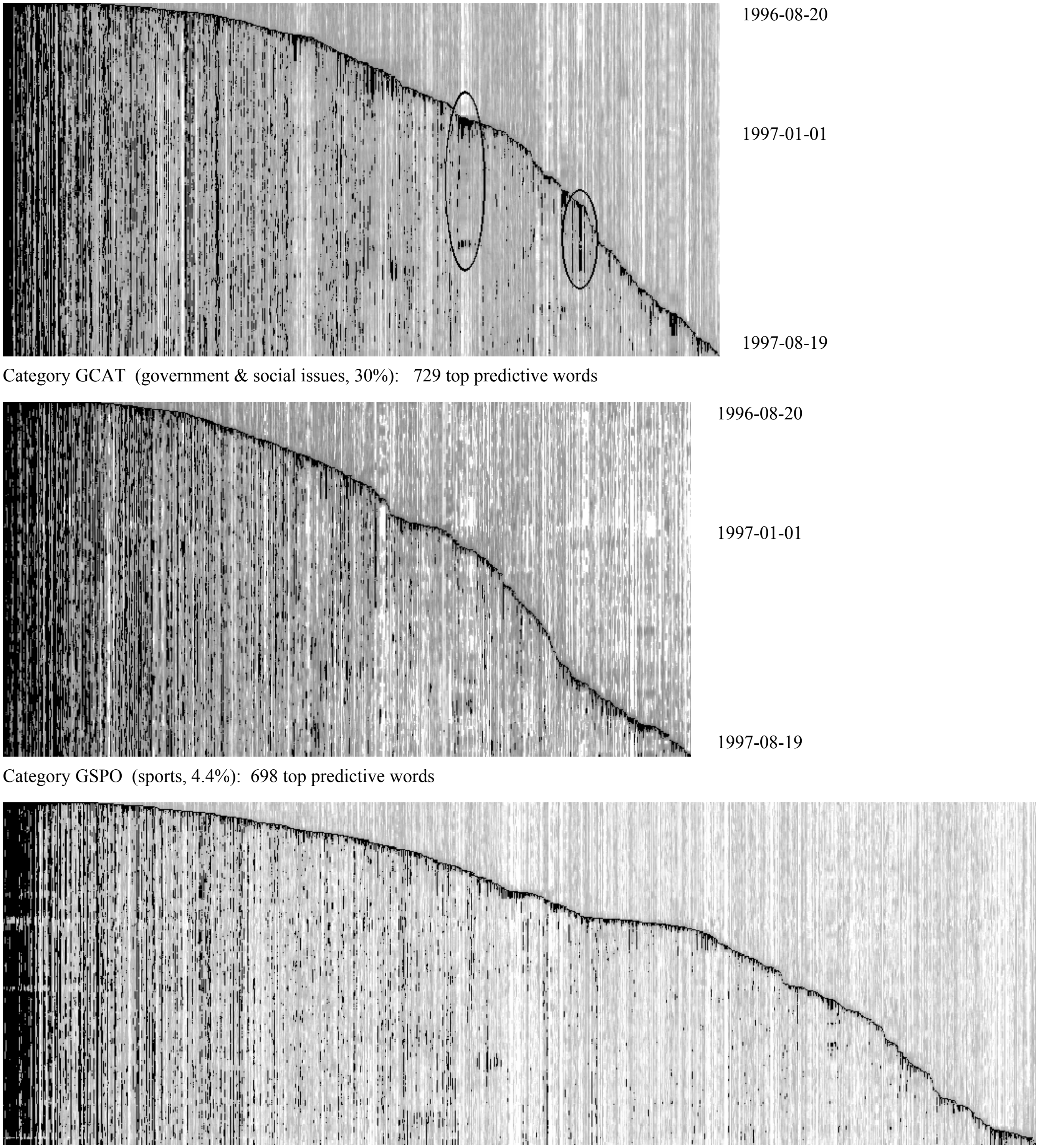
Because of the highly dynamic nature of news streams and other industrial datasets, a classifier built from training cases up to day 30 is unlikely to be effective on day 60 when the top predictive words have mostly changed. To cope with this problem, any operational setting must provide an ongoing stream of additional labeled training cases, but hopefully need as few as possible.

4. Daily Classification Task (DCT)

Concept drift is admittedly a difficult research area. To promote its study, we define a conducive problem formulation we call the *Daily Classification Task*. In it, time is discretized into periods, e.g. days, and of the many cases each day, a limited size iid random sample is provided as a labeled training set, as in [13]. A performance objective of interest, such as classification accuracy or F-measure, is computed on each day of a benchmark dataset, and the average is reported over all days. Using *all* days gives a natural preference for methods that improve quickly with only a few past days available, as opposed to beginning the average after day 100, for example, when steady state may have been achieved. For research purposes, the size T of the daily training set should be selected so that the learning curve is still climbing.

As an optional variant, some percentage *h* of the ground truth test labels may be revealed for *past* days. We call this variant setting *hindsight DCT*—a reasonable assumption for certain real-world settings. For example, in the Reuters setting, some of the predictions that were wrong may get noticed and corrected by people after the prediction errors have been committed. In some settings there is no cost to obtain past labels. This is common where the classification task constitutes a prediction about the future, e.g. whether now is a good time to spin down the laptop disk drive to spare the battery; after the fact, we can determine from disk demand whether the prediction was good.

The strawman learning model is simply to train a state-of-the-art classifier on each daily training set of size T, and then use it to



Category GCAT (government & social issues, 30%): 729 top predictive words

Category GSPO (sports, 4.4%): 698 top predictive words

Category ECAT (economics, 15%): 1081 top predictive words

Figure 1. FxTime visualization of predictive features (columns), revealing recurring predictive features over time (downward).

classify the rest of that day's cases. To surpass this strawman, we would like to leverage the available past training data somehow. An obvious idea is to use a sliding window that retains data from the most recent P previous days. Our empirical evaluation ahead refutes this popular idea for Reuters classification, because of the rampant concept drift from day to day.

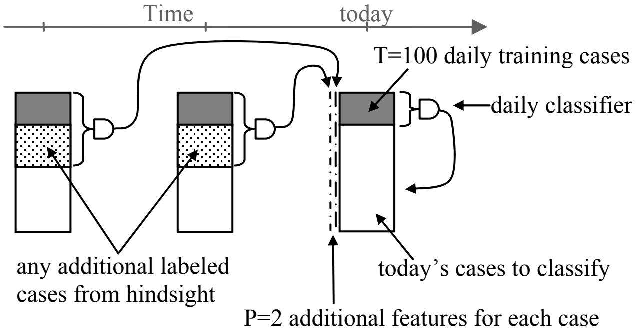

Time
today
T=100 daily training cases
daily classifier
any additional labeled cases from hindsight
today's cases to classify
P=2 additional features for each case

Figure 2. Temporal Inductive Transfer (TIX) Model

5. Temporal Inductive Transfer (TIX) Model

Ideally we would like to be able to leverage the learned *models* from the past, not just reuse past data. This is the research area of inductive transfer, which has met with limited success. Here we desire to apply it in a new way: temporally within a single, changing classification task to help cope with its concept drift.

We propose the following method (refer to Figure 2): Like the strawman, each day we learn a new classifier from the T training examples using whatever state-of-the-art induction algorithm is available. But the input feature vector, in addition to the usual bag-of-words features, we augment with P additional binary features. These are generated by the predictions that the P previous daily classifiers would have made for today's cases. These predictions are computed both for the training data and for today's testing data. Clearly this can be done without knowing the ground truth labels for the testing data.

These additional P features may be potentially predictive in today's daily learning task. If a news theme recurs that was popular within P days ago, the new classifier may be able to leverage the predictions made by the old classifiers that were trained while the theme was previously popular. This suggests some pressure to maximize P for greater long-term memory, if we can afford the computational cost. But note, old classifiers are just reapplied, which is much faster than their original training. Even if we were to let P go as high as 364 days, it is still dwarfed by the large number of bag-of-words features generated by the training sets.

Now, if some or all of the P features end up being worthless with regard to the daily learning task, then the state-of-the-art classifier will be able to ignore them, just as today's text classifiers are easily able to ignore a large number of non-predictive words. Hence, we expect that these P features will not make the text classifier perform any worse, but sometimes may help it improve.

A detail: if P=1, then today's classifier depends on yesterday's classifier only. But that classifier depends on the one from the day before it, and so on. The recurrence relation implies that all classifiers remain in use for all time. Intelligent pruning may someday be devised, but for the purposes of this paper, we break the recurrence by simply substituting a classifier trained on yesterday's data, having no additional TIX features. Hence, by this tweak, the P classifiers operate independently of one another.

6. Empirical Evaluation

We conducted a series of Daily Classification Task experiments on Reuters for each of the four classes: GCAT, GSPO, ECAT and M13 (money markets, 7% positives). That is, for GCAT, we focused on the binary classification task of predicting which of each day's news articles belong to the GCAT topic (government/social, ~30%). Each day we made available T=100 training cases. Rather than the overkill of testing on the many thousands remaining each day, we considered only the first 400 articles each day, selecting the T training cases from these at random. We report F-measure macro-averaged over all days.

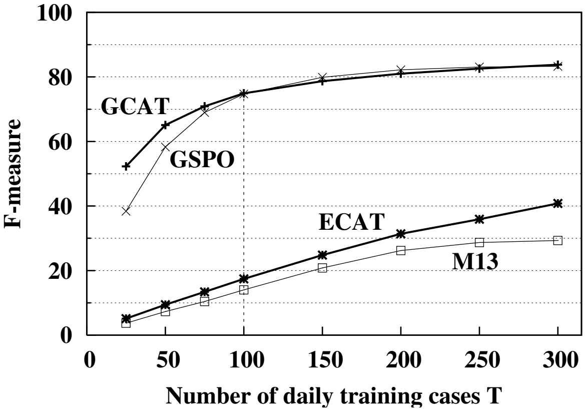


Figure 3. Learning curves for strawman SVM.

The base classifier we used is a linear support vector machine (SVM) trained on binary bag-of-words features (title+body text lowercased, alpha only, max 50K words from each training set), as implemented by the WEKA v3.4 library [19], with BNS feature scaling [4]. We chose this classifier for its state-of-the-art performance and for its ability to tolerate many useless features, to which TIX may add some useless features. For the TIX model, the added features are each binary predictions (preliminary tests with real-valued score features showed worse results).

6.1 Results

Figure 3 shows the average F-measure for strawman, i.e. simply training on the T random samples each day and testing on the rest of the day's cases. This establishes a baseline F-measure performance for T=100, used hereafter. This graph confirms for each class that the choice of T=100 is sufficient for *some* learning to occur, but not so much that additional training data or predictive features would provide no benefit.

Figure 4 contains the four graphs corresponding to the independent experiments on the four Reuters classes. The top two graphs share a common y-axis scale, but the bottom two share a much lower F-measure scale. We are less concerned about absolute performance for each task than about improvement.

Each graph shows the average F-measure performance of all models for T=100 daily training cases. The leftmost point shows strawman, which leverages none of the past training data available. The sliding window technique (labeled Window) adds P previous days of training cases to the daily training set, i.e. the training set grows to T*(1+P) cases. As we increase P, its performance consistently declines, even below the visible chart. For larger P, we see that increasingly stale training data misleads the classifier badly about what the concept today is. Hindsight would only worsen this. Although sliding window is not a viable method for highly drifting concepts, the shape of its performance decline curve might be used as a way to characterize the pace of concept drift in a dataset.

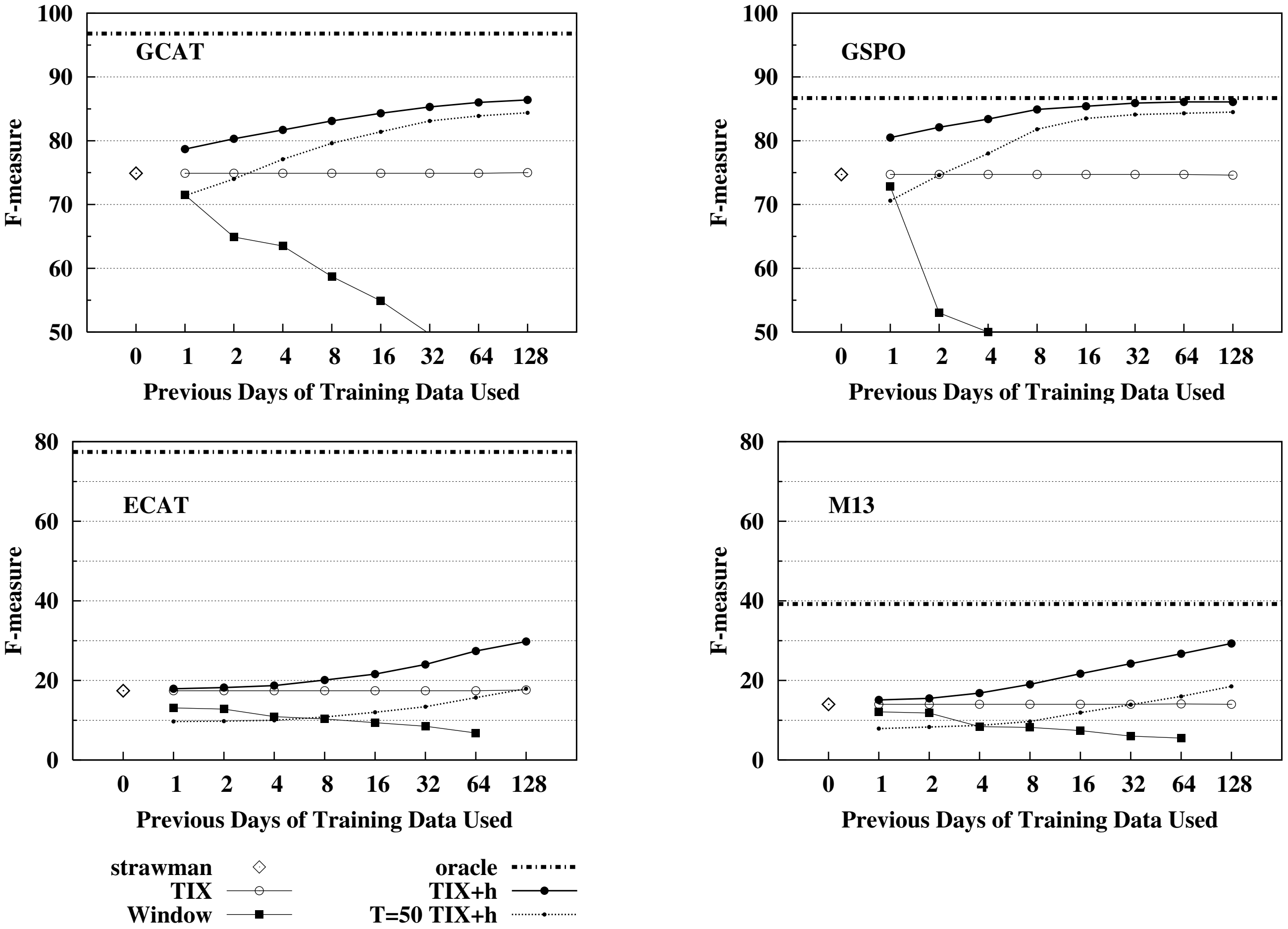


Figure 4. Results for GCAT and GSPO (top), ECAT and M13 (bottom).

Turning now to the TIX model, with T training cases and no additional training labels revealed from the past (*h*=0%), it performs exactly equal to the strawman, regardless of P. One might conclude that the predicted classes from past classifiers have no bearing on the daily learning task at hand. But this is not so. If we increase the training sets of those P past classifiers by revealing all *past* training labels (*h*=100%), then they are able to generate predictive features that are more accurate, and they indeed become useful to the daily learning task (see the climbing curve labeled TIX+h). We observe a consistent and substantial rise of more than 10 points of F-measure for all four classes as we increase P to 128 past days. (We repeated this measurement for 30 of the most common Reuters topics: from CCAT at ~47% prevalence down to C11 at ~3%, roughly correlating with classification difficulty. TIX+h at P=128 shows a substantial gain for all but the most difficult topics, as shown in Figure 5. The classes are arranged along the x-axis according to TIX performance. We return our attention to Figure 4 hereafter.)

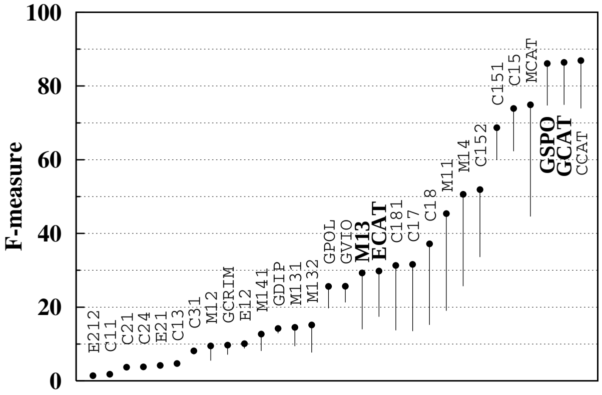


Figure 5. TIX (dot) vs. strawman (whisker) for 30 classes.

Oracle: For an upper-bound performance comparison, we also evaluated an 'oracle' model: we train the daily classifier on only the T=100 training cases, like the strawman, but we include one additional binary feature that gives away the true class label for each case. The learned classifier is not perfect, since SVM does not memorize its training set, as k-Nearest-Neighbors would. But this gives an upper bound on the performance we could expect from the base classifier we used, if the TIX predictive features had been 100% perfect.

In the case of GSPO (top right), TIX+h rises to match the performance of the oracle model. For GCAT and M13, TIX+h rises over half way to the oracle from strawman performance.

Varying daily training size T: For GCAT and GSPO, we nearly match the performance of TIX+h even with *half* as many training cases each day (see curve labeled T=50 TIX+h), at least when

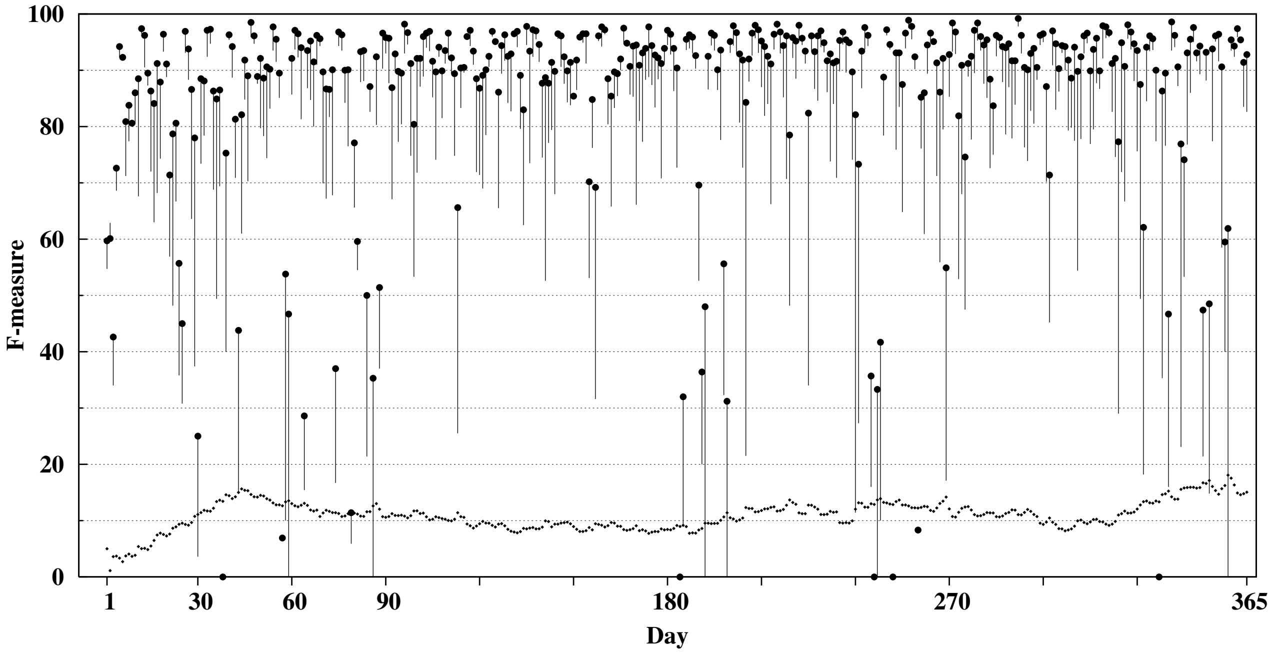


Figure 6. F-measure improvement by TIX with hindsight (dot) over strawman (end of whisker) for each day of GCAT. The 30-day moving average of differences is shown by the dotted line near the bottom, averaging +10 F-measure points.

P=128 past days of hindsight memory are allowed. This suggests the TIX features we add are worth nearly as much as doubling the number of rows in the training set. For some real-world settings, halving T could cut in half a substantial daily manual effort in labeling cases.

In contrast, for ECAT and M13 where the overall classification accuracy is much worse, TIX+h with T=50 never approaches the performance with T=100. One might suppose that the P TIX classifiers have difficulty in predicting accurately, however, with h=100% hindsight, they may be getting roughly the accuracy of the oracle model. For ECAT this is quite high, so rather than blame the P=128 TIX features for being inaccurate, instead we may reason that their applicability to the new daily classification task is less than for classes such as GCAT or GSPO.

In these difficult situations, there is little substitute for fresh training data, though even here more days of memory helps monotonically. If we had a dataset that covered a longer time period, it would be interesting whether P=1000 or more could eventually bring the performance up to that of the oracle.

6.2 Time Series View

Drilling down on the GCAT results for T=100, Figure 6 shows the performance improvement for each day of the year. The dot indicates the F-measure of TIX with hindsight using P=128 days of memory, and the other end of the whisker indicates strawman. The 30-day moving average of the improvement is shown by the dotted line at the bottom, averaging about +10 F-measure points. Also notice that when strawman performance is very low or even failing, TIX using hindsight often led to large improvements.

What is more striking is that all the differences are positive (with the minor exception of a small loss that occurred on day two). This substantiates our claim that the daily induction task can leverage TIX features when they are useful and successfully ignore them when they are irrelevant. This is a property of the base classifier we have chosen. If we had used a base classifier that was very sensitive to feature selection, such as Naïve Bayes, then we would expect to see some losses as well.

6.3 Reduced Hindsight

So far, we have only considered full hindsight or no hindsight. Figure 7 shows for each category, the F-measure of TIX as we vary hindsight: 0%, 25%, 50%, and 100% of past test case labels revealed. Recall that at 0%, its performance happens to match strawman. Each day there are 400 articles; T=100 are used for training, leaving 300 for testing. At 25% hindsight, 75 of the test cases later have their true labels revealed, and the old daily classifiers are retrained for their use as feature generators. For GCAT and GSPO, 25% hindsight yields most of the benefit of full hindsight. This non-linear behavior is typical of learning curves, and practically useful in many real-world domains where there continues to be some cost for obtaining hindsight labels. But for the difficult class ECAT, 25% hindsight gives no benefit.

6.4 Runtime Analysis

The size of the available training data grows linearly over time. For the sliding window algorithm, it can accumulate a very large training set. For an induction algorithm such as SVM, this results in an O(n^2) training time. In practice, we do not see worst case performance. We found consistently that doubling P results in tripling the time to run the sliding window experiment with no hindsight. In contrast, doubling P for TIX *with 100% hindsight* only increases the time by 1.7x. Concretely, sliding window for P=32 (3300 training cases) took ~16 hours on modern hardware to complete the experiment, while TIX with 100% hindsight took ~2.7 hours, which effectively leverages 12,900 training cases. We did not even attempt sliding window with hindsight: besides slowing down the training time tremendously, we reason that the

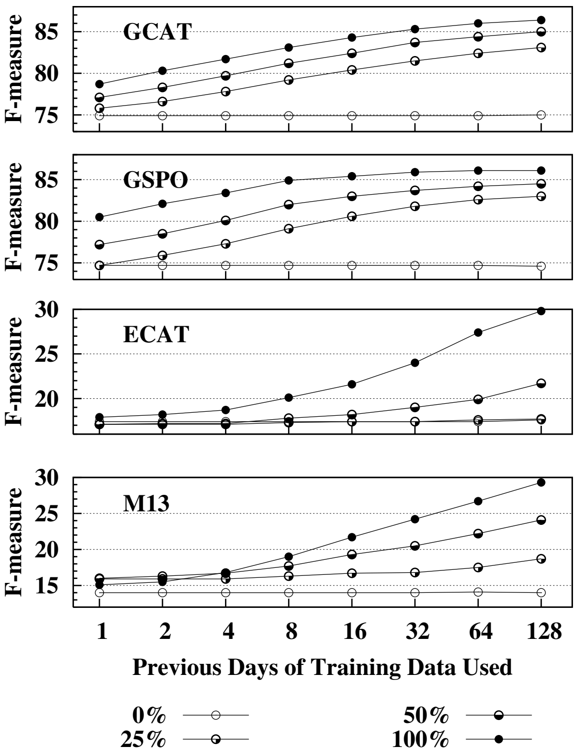


Figure 7. F-measure for TIX, varying % hindsight.

additional stale cases would only further drown out the signal from the T=100 fresh daily training cases.

The secret to the efficiency of TIX is that it brings forward its *induced* linear classifiers for future uses, which are quick to evaluate on new cases. It need not save or re-train on past data. It trains once for T=100, and once more for the hindsight cases. Hence, its runtime is only linear in P.

7. Discussion

It is somewhat disappointing that the TIX model provided no benefit without hindsight. We believe the reason is that its inductive transfer features have no greater accuracy than what the induction algorithm can already learn from the bag-of-words features. (An alternate hypothesis is that the daily classifier *does* learn to depend on the TIX features, but their unreliability results in no performance gain. But by painstakingly examining the feature weights on the TIX features, we found they were not valued.) With some additional hindsight data, they then obtain greater accuracy and become useful in the daily learning task, even though they may be for a somewhat drifted concept.

Recall that for this study, each past TIX classifier did not build on previous classifiers, in order to break the recurrence chain to the first day. If they had instead each leveraged previous classifiers, one possibility is that they would then perform more accurately, and hence eventually provide more reliable features for future daily classifiers. In this case, TIX without hindsight might surpass strawman. We are skeptical of this: the outputs of strawman performed equal to those of the P=1 TIX classifier, i.e. there would presumably be no difference in performance if we were to replace the past strawman classifiers in TIX with the daily TIX classifiers that are trained with the additional P=1 feature. Nonetheless, this idea should be tried, and if indeed successful, some intelligent pruning method will eventually be required to avoid linear slowdown over thousands of days' data.

Regarding sliding windows: In additional experiments not recorded here, we tried sliding window for training sizes T=10…100. As expected, the greatest effect of reducing T is a loss in accuracy. But a minor effect is that for very small training sets, expanding the window to include a few past days of data can be beneficial. For example, for GCAT with T=25, sliding window peaked at 57% F-measure with P=4, up from 53% for strawman.

Finally, although our results discourage sliding window, we acknowledge that in settings without the time discretization of the Daily Classification Task formulation, some sort of sliding window scheme may be needed to select a sample of recent data. Our RCV1 results suggest strong pressure to minimize the window width, and instead use TIX to leverage older data.

8. Related Work

In most settings in the concept drift literature, concepts change rarely or else gradually. This has led to many heuristic methods to detect *when* a significant change occurs. But this is a non-issue in the Reuters data and industrial datasets of interest where substantial change is the rule, rather than the exception.

The remaining research challenge is how to produce an effective classifier despite the concept having just drifted. The prevailing approach in the literature is to completely throw away the old classifier and most of its training data, and then build a new one on more recent data (e.g. CVFDT [10] and a variety of sliding window techniques, some with adaptive window sizes [18][13]). Two approaches stand out by their ability to re-use older information. One selects various old training sets for inclusion, if they appear to be similar to the most current training data [13]. The other reuses previously learned concepts, rather than re-using the training data they represent. Examples include reactivating a single old model, if it seems more appropriate on recent data than the current classifier (e.g. FLORA3 [18]), while other models use ensembles of old classifiers and prune or adapt the weights according to recent data [16]. A direct empirical comparison would be interesting; though in none of these approaches does a previously learned classifier benefit the training of a new base classifier, as in TIX.

There has been a great deal of research on inductive transfer under many names, e.g. multi-task learning, lifelong learning, bias learning, representation learning, and notably Hierarchical Bayes. These efforts show consistently that transferring knowledge helps from 'similar enough' task data, but if the related task is 'too dissimilar' it hurts (politely called 'negative transfer'). This was one of the greatest concerns voiced at the recent NIPS workshop on inductive transfer [17]. By contrast, it is noteworthy that the TIX model never harms prediction accuracy. This property is designed into the model by the ability of the base SVM classifier to successfully ignore useless features. This is its great strength.

Another difference is that many existing approaches to inductive transfer do not actually transfer previously induced models, but instead re-use the old training data to help condition the induction of new classifiers. This requires ever more CPU time for retraining on growing training sets, which unfortunately is super-linear.

For text learning, there has been a great deal of experimentation with different feature vectors. Besides the many variants that try stemming, phrases, and other linguistic techniques, some replace the feature vector with a representation thought to model the dataset better, e.g. Latent Semantic Indexing, distributional clustering, and cluster centroids (e.g. [1][8][9]). Some of these methods have the advantage that they can leverage large bodies of unlabeled text—semi-supervised learning. But again, if the unlabeled data are 'too dissimilar' then the changed representation may instead defeat the learning task at hand.

Some related work uses the output of classifiers as features, e.g. stacking, voting, and various ensemble methods. These methods all train their subclassifiers on the *same* input training set. *Sequential prediction* methods use the output of classifiers trained with previous, overlapping subsequences of items, assuming some predictive value from *adjacent* cases, as in language modeling.

9. CONCLUSION

We have shown the success of temporal inductive transfer for the DCT setting when the ground truth labels for some past test cases are eventually revealed—hindsight. While useful in many real-world situations, in others the past labels are not available without additional expense. Thus, further research is called for in the DCT setting without hindsight.

A promising avenue for future work includes hybridizing the temporal inductive transfer idea with related work in semi-supervised learning. The past labeled data provides for a richer setting than traditional semi-supervised learning. Interestingly, Gabrilovich and Markovitch recently tried augmenting the bag-of-words feature vector with the output of classifiers trained on the Open Directory hierarchy, and found some benefit [8]. While their inductive transfer is not through time and not from markedly similar tasks, the benefit of augmenting vs. replacing the raw features we believe is the right approach.

10. ACKNOWLEDGMENTS

Ian Witten's WEKA software library for machine learning [19] made this research a pleasure—only 200 lines of code. We thank Eric Anderson and the HP Labs Utility Datacenter for providing ample computing horsepower to meet the conference deadline.

11. REFERENCES

[1] Baker, L. D. and McCallum, A. K. Distributional clustering of words for text classification. In *Proc. of the 21st Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR, Melbourne), 1998.

[2] Fawcett, T. ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. Hewlett-Packard Labs, Tech Report HPL-2003-4, 2003. See http://www.hpl.hp.com/techreports/2003

[3] Fawcett, T. and Flach, P. A response to Webb and Ting's 'On the application of ROC analysis to predict classification performance under varying class distributions.' *Machine Learning*, 58(1):33-38, 2005.

[4] Forman, G. BNS Scaling: A Complement to Feature Selection for SVM Text Classification. Hewlett-Packard Labs technical report, HPL-2006-19, 2006.

[5] Forman, G. Quantifying Trends Accurately Despite Classifier Error and Class Imbalance. Submitted to the *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD, Philadelphia), 2006.

[6] Forman, G. Counting Positives Accurately Despite Inaccurate Classification. In *Proc. of the European Conf. on Machine learning* (ECML, Porto):564-575, 2005.

[7] Forman, G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, *Special Issue on Variable and Feature Selection,* 3(Mar):1289-1305, 2003.

[8] Gabrilovich, E., and Markovitch, S. Feature Generation for Text Categorization Using World Knowledge. In *Proc. of the 19th Intl. Joint Conference for Artificial Intelligence* (IJCAI, Edinburgh), 2005.

[9] Han, E. and Karypis, G. Centroid-Based Document Classification: Analysis & Experimental Results. In *Proc. of the 4th European Conf. on the Principles of Data Mining and Knowledge Discovery* (PKDD): 424-431, 2000.

[10] Hulten, G., Spencer, L., and Domingos, P. Mining time-changing data streams. In *Proc. of the 7th Int'l. Conf. on Knowledge Discovery and Data Mining* (KDD, San Francisco):97-106, 2001.

[11] Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proc. of the 10th European Conf. on Machine Learning* (ECML, Berlin):137-142, 1998.

[12] Karypis, G. and Han, E. Fast supervised dimensionality reduction algorithm with applications to document categorization & retrieval. In Proc. of the 9th Intl. Conf. on Information and Knowledge Mgmt (CIKM):12-19. 2000.

[13] Klinkenberg, R. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, *Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift,* 8(3):281-300, 2004.

[14] Lewis, D., Yang, Y., Rose, T., and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. *J. of Machine Learning Research*, 5(Apr):361-397, 2004.

[15] National Institute of Standards and Technology (NIST) Reuters Distribution, http://trec.nist.gov/data/reuters Also: http://about.reuters.com/researchandstandards/corpus

[16] Scholz, M. and Klinkenberg, R. An Ensemble Classifier for Drifting Concepts. In *2nd Int'l. Workshop on Knowledge Discovery in Data Streams*, (ECML,Porto):53-64, 2005.

[17] Silver, D., Bakir, G., Bennett, K., Caruana, R., Pontil, M., Russell, S., Tadepalli, P., organizers. Workshop on Inductive Transfer: 10 Years Later. *19th Conf. on Neural Information Processing Systems* (NIPS), Dec. 9, 2005.

[18] Widmer, G., Kubat, M. Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, 23(1):69-101, 1996.

[19] Witten, I. and Frank, E., *Data mining: Practical machine learning tools and techniques (2nd edition),* Morgan Kaufmann, San Francisco, CA, 2005.