ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200271

Minimizing and Recovering from the Effect of Concept Drift via Feature Selection

Daegun Won and **Peter Jansen** and **Jaime Carbonell**[1]

Abstract. With increasing expectations for flexibility and adaptability of machine learning systems, the importance of automatic model updates and performance stability in the face of various types of concept drift has received significant interest. In this study, we explore how feature selection techniques, mostly neglected in the aforementioned effort, may be used to improve the drift compensation process with no a priori assumptions regarding the type of drift. To this end, we (A) evaluate several feature selection techniques by their potential to minimize the effect of drift while still capturing its essence (predict its near-term course), (B) analyze the factors contributing to the success of our proposed method, and (C) provide empirical drift adaptation results via active learning on an extensive data set of real-life political indicators. The results demonstrate that using L1 regularization in the context of our new sample-driven drift-modeling approach results in improved performance as compared to alternative feature selection techniques. The reduced model also requires fewer additional samples to recover from drift even with existing active-sampling strategies.

1 INTRODUCTION

With growing applications and increasing expected lifetime of machine learning systems, the importance of automatic model adaptability (performance stability) in the face of various types of concept drift has been getting significant interest. In this study, we explore how feature selection/reduction techniques may be used to enhance or streamline drift detection and compensation algorithms.

We present a technique composed of feature selection in combination with a drift adaptation system consisting of a (fairly naive) model transfer with subsequent active learning for performance recovery (i.e., a system similar to the one described in [47]).

In section 2 we try to situate our effort in the context of related work on feature selection, feature drift and drift adaptation. In section 3 we examine parameter-based drift modelling and show how it naturally extends to our approach of integrated feature selection. Section 4 describes the drift adaptation system (framework) that we used for our current evaluation. Section 5 deals with the details of our experimental setup, including data, metrics and parameters. Section 6 presents a discussion of the results, and in Section 7 we conclude and look ahead at future work.

2 RELATED WORK

There is a vast literature on *feature selection*, and hundreds of algorithms have been proposed [5, 6, 16, 22, 37, 39, 49] etc. Such algorithms may be distinguished based on how they generate candidate feature (sub)sets, how they evaluate them, how they decide on relevance, and how they verify validity of the result [11]. In the vast majority of cases, the objective is to avoid the curse of dimensionality (leading to overfitting because of low feature space coverage of the training set) and to make the classification task computationally manageable. The goal is to remove all irrelevant features and keep all the relevant features needed for the (classification) task at hand. However, neither overfitting nor computational load are primary concerns for the data sets we have worked with.

The phenomenon of *concept drift*, too, has recently received enormous amounts of attention, mainly due to the need for online classification of very large, constantly changing data streams [1, 4, 26] etc. The field itself has been subject to drift, going from detection and characterization [15, 14, 1] etc., via classifier retraining towards anticipation and proactive adaptation [47, 8, 12].

For the purpose of the current work, we are in fact interested in the fastest possible recovery from drift, in particular how this relates to feature selection, a factor not previously considered in this context.

More related to our work and less studied than the above, is the area of *feature drift* [3, 32, 34], i.e., the concept drift resulting from the changing relevance of individual features over time in a given classification task. This is seen as a problem in itself which necessitates incremental or wholesale dynamic retraining of a classifier. Instead, our goal is to examine to what extent the choice of feature set may influence the drift sensitivity and recovery of an overall drift adaptation system, in which the drift may or may not be the consequence of feature drift per se.

As far as we know, the sample-driven drift modeling described in section 4, where features are selected based on drift characteristics and past history, has not been tried in the literature. As such, we used classic feature selection methods such as Chi-Square, ANOVA, and MI for comparison using the same system and time window.

3 CHARACTERIZING DRIFT WITH MODEL PARAMETERS

3.1 Traditional parameter-based drift-modeling

In a supervised learning setting, a mapping function $f : X \to Y$ is chosen from the hypothesis space F such that f is a close approximation of the true hypothesis $g : X \to Y$ given the observed data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Note that g may or may not be in F. The selected hypothesis f can be defined with the model parameters θ.

When the environment is non-stationary and subject to drift of a significant degree, the true hypothesis g_t at time t may change substantially over time. Consequently, in order to remain an accurate

[1] Carnegie Mellon University, USA, email: {daegunw,pjj,jgc}@cs.cmu.edu

representation, the model f_t needs be re-trained or adjusted with new data to produce updated parameters θ_t for each time t.

Assuming F includes functions $\{f_t\}$ that can approximate $\{g_t\}$ sufficiently well, we can describe the drift as a regression model $\pi : T \rightarrow \Theta$, where $\pi(t) = \theta_t$ and θ_t represents the model parameters of f_t. When π can be approximated with a certain functional form $\hat{\Pi} : T \rightarrow F$, we can train $\hat{\pi}$ with $\{\theta_1, \theta_2, \ldots, \theta_t\}$ to produce $\hat{\pi}(t+1) \approx \theta_{t+1}$. For example, a (temporally) linear drift would be in the form of $\pi(t) = \theta_0 + \delta_\theta t$ parameterized with θ_0 and δ_θ. More formally, given $\{\theta_0, \theta_1, \cdots, \theta_T\}$, π would be obtained via the following optimization:

$$\hat{\pi} = argmin_\pi \sum_t L(\pi(t), \theta_t) \tag{1}$$

where $L(\pi(t), \theta_t)$ is a loss function of choice such as a quadratic loss function.

3.2 Issues with traditional parameter-based drift modeling

While the idea of using model parameters to model the drift seems efficient as long as the hypothesis space F includes an approximation of ground truth, estimating π does involve several problems in practice. One important problem is the *high cost* of maintaining an accurate drift model. To make an accurate estimate $\hat{\pi}$, each set of estimated model parameters θ_t must be accurate to reduce model bias. Also, the number of estimated parameters should be large enough to reduce model variance. For logistic regression, SVM or any rotationally invariant classifier, $\Omega(m)$ samples would be required in the worst case to estimate each θ_t, where m is the number of features [39]. For systems requiring high reliability, the drift model would need to be updated frequently to ensure performance, compounding the cost. Frequent updates also limit the number of obtainable new samples, as acquiring new labels takes time. As a result, performing an $\Omega(m)$ operation frequently is not only just costly, but sometimes also simply physically impossible.

Another difficulty results from *large feature sets*. It is a common practice in recent machine learning applications to incorporate as many features as available so as to maximize the amount of accessible information. While a large set of potentially redundant features can be helpful in maximizing predictive performance, it may also complicate meta-analysis of the task such as needed for human understanding or, in our situation, *drift analysis*. Redundant features introduce a large number of viable alternative hypotheses $F_t = \{f'_t \in F : f'_t \approx f_t\}$ with reasonably similar capability. As a simple example, consider model parameters $\theta_t^{(i)}, \theta_t^{(j)}$ for two *identical* features at time t. Assuming no constraints or regularization of parameters, any values of $\theta_t^{(i)}, \theta_t^{(j)}$ would be acceptable as long as their sum remains the same. This means that the change of each individual parameter ($\theta_{t+1}^{(i)} - \theta_t^{(i)}$) can be arbitrarily large, and relying on such changes is therefore not likely hideRelying on parameters of a single hypothesis at each time t in such situations is not likely to produce an accurate drift model. In order to correctly model the drift in a such case, one would need to identify the suitable series of hypotheses over $F_1 \times F_2 \times \cdots \times F_T$ with temporal integrity. Unfortunately, this becomes exponentially challenging as $|F_t|$ or T increases.

The above problems become even more pronounced when a good functional approximate of π is not known in advance. One way to approach the issue is by using simple functions to generate a *local approximation* of the drift. Depending on the general characteristics of drift such as smoothness, a linear projection $\hat{\pi}(t) = \theta_0 + \delta_\theta t$ or an identity projection $\hat{\pi}(t+1) = \theta_t$ may be used. However it is not clear whether the past parameters $\{\theta_1, \theta_2, \cdots, \theta_t\}$ are optimal with respect to the inaptness of the simple projection function. For example, the identity projection might fare better without weakly predictive features with high temporal variance since simply reusing such parameters would just diminish the performance.

3.3 Our approach: sample-driven drift modeling with feature selection

The problems described in the previous section above can be summarized as (1) a high cost, and (2) the difficulty to obtain reliable θ_t's (3) that fit the drift model well. The difficulty of finding reliable θ_t stems from the cascade optimization of drift model, where the previous θ_t's are computed independently without any consideration of the form of the drift model. To cope with these problems, we propose here the following *sample-driven drift modeling with feature selection*. Using a reduced set of features may in addition result in a drift model with fewer parameters and improve responsiveness of the system against drift by focusing on a more reliable or stable set of features.

In Equation 1, the loss function $L(\pi(t), \theta_t)$ depends on the estimated parameters θ_t. Although this may be an effective approach when data storage or computational power is limited, we can instead use the labeled data itself; $L(\pi(t); D_t)$. Given the modified loss function L, we can then optimize a sparse projection function π using the following form of objective function extending Equation 1:

$$\arg\min_\pi \sum_t L(\pi(t); D_t^l) + \lambda \sum_t \phi(\pi(t)) \tag{2}$$

where ϕ is a penalty term that promotes sparsity of π and the D_t^l is the set of labeled samples obtained at time t.

4 PROPOSED FRAMEWORK

Algorithm 1 describes our active drift compensation framework. It involves two major components: (1) the feature selection for the drift model, and (2) an adjustment to the new environment via active learning and transfer learning. We used a Logistic Regression model as our target classifier, but in principle any classifier may be used without any significant change. In the following subsections, we describe these components in more detail.

4.1 Feature selection for the drift model

Equation 2 provides a general form where drift modeling and feature selection can be jointly optimized. In this work, we focus on the most simplistic drift modeling that requires no prior knowledge about the drift, the identity projection $\pi(t) = \theta_{t-1}$. We hypothesize that the significantly relevant set of features with respect to the drift is locally stable. Note that this assumption is different, and often more relaxed than the smoothness assumption ($\|\theta_{t+1} - \theta_t\|^2 < \delta_{max\_drift}$). While the smoothness assumption limits the magnitude of drift by blindly posing an upper bound to the change in the model parameters, our assumption does not restrain the magnitude of drift.

Given that we are using an identity projection to extrapolate the drift, we can substitute $\pi(t) = \theta_{t-1}$ with θ in equation 2:

$$\begin{aligned} &\min_{\pi} \sum_{t \in \mathrm{T}} L(\pi(t) = \theta_{t-1}; D_t^l) + \lambda \sum_t \phi(\pi(t)) \\ &= \min_{\pi} \sum_t L(\theta; D_t^l) + \lambda \sum_t \phi(\theta) \\ &= \min_{\theta} \sum_t L(\theta; D_t^l) + T\lambda\phi(\theta) \end{aligned} \quad (3)$$

The penalty term then depends on a single set of parameters θ, allowing the use of classic feature selection penalty terms such as L_1 norm without losing convexity. It is important to note that with inaccurate drift projections such as the identity projection the resulting model θ does not result in a good performance at each time t and the nonzero parameters are used purely for feature selection only. Due to the use of inaccurate π and θ, the window size $|\mathrm{T}| = w_f$ should be set small. In order to find a subset of important features over T while maintaining the predictive power of the classifier as much as possible, we employ *L1 regularization*, which minimizes the following:

$$\arg\min_{\theta} \sum_{t' \in [t-w_f, t-1]} l(\theta; D_{t'}^l) + \lambda|\theta|$$

The objective function offers a trade-off between predictive performance over the time window (i.e., the predictive power with respect to drift) and the number of features used. The resulting non-zero features are selected and used to train and update the target classifier.

Since there is, to the best of our knowledge, no related prior work, we have also tested classic feature selection methods such as chi-square, ANOVA, and mutual information for comparison, using the same time window.

4.2 Drift Compensation

Once a subset of features is selected from the previous component, we try to recover from the environment change by (1) transferring the knowledge available from the previous time epochs and (2) using active learning to select a small number of samples to be labeled. To transfer the knowledge between the past epochs and the current time t, we simply reused the past labeled samples along with the newly acquired samples, which is equivalent to the following:

$$\theta_t = \arg\max_{\theta} \sum_{t' \in [t-w_{tr}, t-1]} l(\theta; D_{t'}^l) + l(\theta; D_t^l)$$

It is worth noting that our use of this rather simple transfer method was largely in part to cope against the scarcity of the positive examples and produce good generalization of the task. When using balanced data, regularization based transfer or other methods may be used.

For active learning strategy, we limited ourselves to a simple uncertainty sampling that chooses an instance with the highest label entropy. Although it is one of the most popular existing methods in many applications, it does not explicitly exploit the selected feature set. A more sophisticated active learning or transfer learning technique could be used instead for better efficiency without any additional change to the framework.

Algorithm 1 ACTIVE DRIFT COMPENSATION WITH FEATURE SELECTION

Input: Drifting data $D^t = \{(x_i^t, y_i^t)\}|_{t \in (0, t_{current})}$
Target classifier C
Parameter: Training / feature selection window size w_{tr}, w_f
Sampling iterations N_s, Labeling budget b
Num. of features k

for $t \leftarrow 0$ to $t_{current}$ **do**
  # Feature Selection
  Select k features using $\bigcup_{\tau \in [t-w_f, t-1]} D_l^\tau$
  Let $S_k^t : \mathbb{X} \to \mathbb{R}^k$ be the resulting feature extractor

  # Active Learning
  Initialize $D_l^t = (X_l^t, Y_l^t)$ with a few randomly chosen instances from $D^t = \{(x_i^t, y_i^t)\}$
  $D_u^t \leftarrow D^t \setminus D_l^t$
  $\tilde{D}_l = (\tilde{X}_l, \tilde{Y}_l) = \bigcup_{\tau \in [t-w_{tr}, t-1]} D_l^\tau$
  for $i \leftarrow 0$ to N_s **do**
    Train C_t with $(S_k^t(\tilde{X}_l \cup X_l^t), \tilde{Y}_l \cup Y_l^t)$
    Sample b instances $B_i = \{(x, y) \in D_u^t\}$ given C_t, D_u^t
    $D_u^t \leftarrow D_u^t \setminus B_i$
    $D_l^t \leftarrow D_l^t \cup B_i$
  end for
end for=0

5 EXPERIMENTAL SETUP

5.1 Data

The main dataset we used is a set of measurements and aggregations of political indicators produced by the the Integrated Conflict Early Warning System [2] made available via the Harvard Dataverse [3]: "Event data consists of coded interactions between socio-political actors (i.e., cooperative or hostile actions between individuals, groups, sectors and nation states). Events are automatically identified and extracted from news articles by the BBN ACCENT event coder. These events are essentially triples consisting of a source actor, an event type (according to the CAMEO taxonomy of events), and a target actor. Geographical-temporal metadata are also extracted and associated with the relevant events within a news article." (see also [42, 41, 9]).

Each sample is indexed by time (month / year) and state (country), and spans a period of 198 months starting January 2001. We used a semi-proprietary version of the set in which additional data points, derived ffiand other additional features, and reconstructed missing data were supplied, as well as the ground truth of various categories of the political situation in the given country at that time. Each sample contains 559 features and 5 binary class labels, including international and domestic crises (IC, DPC), ethnic and religious violence (ERV), as well as rebellion (REB) and insurgency (INS).

Due to the complexity and volatility of the global political situation, but also due to changes in news reporting and measurement techniques, it offers a rich data set with multiple occurrences of drift with different magnitude and behavior over time.

We resolved for simplicity to treat the dataset as representing 5 separate binary classification tasks. Each task is significantly skewed towards the negative class, approximately 20 to 1, due to the nature of the data. This resulted in only few positive examples each month in the data, making the task difficult to generalize well. We resolved this problem of class imbalance by grouping instances quarterly instead of each month, ensuring a reasonable number of positive samples in the sampling pool.

In addition to the ICEWS data, we tried our method on two other

[2] https://en.wikipedia.org/wiki/Integrated_Conflict_Early_Warning_System
[3] https://dataverse.harvard.edu/dataverse/icews

(much smaller) real data sets, both containing a similar number of features after preprocessing categorical features.

The *airlines dataset* [15] contains flight departure and arrival records. It has several temporal features such as day of the week, flight departure time, and flight length. We used only the day of the week feature as temporal feature, and split the data into two subsets so that the instances can span over a simulated two week period. We treated the other temporal features as categorical variables via hourly binning.

The *spam dataset* [20] was originally intended for streaming setting, hence the only temporal information in the data was the order in which the instances were listed. We split the instances into 5 subsets by its chronological order and considered each subset as one epoch.

5.2 Experimental Details

In our experiments, we used two time windows: one for feature selection w_f, and one for knowledge transfer between epochs w_{tr}. We intentionally set w_{tr} smaller than w_f to allow better adaptability to drift, while still being able to extract a reliable subset of features. For the active learning component, the batch size b and the number of iterations N_s were set allow the resulting model to achieve high performance . Also, to correctly evaluate the highly skewed ICEWS dataset, the F1 score was used instead of accuracy. The details of window size and other data specific parameters are summarized in Table 1.

More relevant for our investigation was the number of features selected during the feature selection phase, and the related hyperparameter λ for L1 regularization, hard-coded to $\lambda = 0.1$. As λ does not uniquely determine the resulting number of features, we used the average number of features throughout the time span as a parameter for the other feature selection methods in order to make a fair comparison.

For the ICEWS dataset, labeled samples were weighted inversely proportional to the frequency of the corresponding class so that the sum of weights for each class is the same.

Finally, all the results for updates for every epoch were averaged, along with 5-fold cross validation.

Table 1: Experimental details for each dataset

Dataset	Original # Feats	w_{tr}	w_f	b	N_s
DPC ERV INS REB IC	559	4 (12mo)	8 (24mo)	5	50
Airlines	634	1	2	50	100
Spam	499	1	2	5	50

5.3 Metrics

To quantitatively evaluate the improvement of drift adaptability, we use the following metrics (as the drift adaptation is done via active learning, the metrics are similar to the commonly used metrics to evaluate active learning strategies):

- *Area under the Learning Curve (ALC)*: This estimates how the feature selection affects the model adaptability to drift throughout its lifetime.
- *Recovery speed*: Rx = Number of batch sampling iterations needed to achieve a (pre-specified) $x\%$ of the highest performance among all 4 methods. This metric directly addresses how fast a model can adapt to environment change.

6 DISCUSSION AND ANALYSIS OF RESULTS

6.1 Drift adaptability by feature selection method on the ICEWS dataset

We can compare the performance of different feature selection methods along three qualitative aspects: (1) a higher *initial point* indicates that the selected features capture properties less susceptible to drift (meaning that labeled data from previous epochs can still be used effectively). Further, (2) a higher *saturation point* indicates a higher representative power of the selected features with respect to the new concept. Finally, (3) the *slope of the learning curve* relates to model size as well as efficiency of the active learning. Figure 3 shows the average F1 recovery curves for several classes for different feature selection methods. For all classes our method resulted in better performance in the first and the last aspect. That the representative power of a reduced feature set is less than the (full-set) baseline is not surprising, but with the exception of the DPC class, the loss was minimal.

Tables 2 and 3 summarize the performance recovery via active learning on all the classes with different feature selection methods. On average, feature selection via L1 regularization resulted in a 1 – 3% increase in ALC and a 50 – 70% decrease in labeling effort to recover most of the performance level compared to the no-feature-selection baseline, except for one case (DPC with 188 features, see Figure 3e). However, performance was greatly improved with a hyperparameter of $\lambda = 10$ (roughly tuned by cross validation), when the model selected 65 features on average (Figure 3f).

A sensitivity analysis (Figure 4) shows the effect of the hyperparameter for two of the classes (DPC and ERV). The results indicate that there is an optimum setting, but that it may be strongly class dependent. Table 3 indicates that the other feature selection methods nearly always failed to achieve the target performance level within the labeling budget. Hence we only report comparison of our method (with L1) with the no-feature-selection baseline.

Somewhat surprising was that, even with an aggressive feature reduction (down to 10 – 25% of the original features) our method resulted in very little performance loss, given enough data. While this could in part be due to the fact that some of the features in our dataset were clearly redundant or irrelevant, this does not explain why only the L1 regularization method was able to take advantage of this. We plan to investigate this further in the future.

6.2 Why is L1 working better on the ICEWS data?

One potential explanation of the success of the L1 regularization can relate to the average absolute pairwise correlation of features. We have computed the average pairwise correlation of features selected over the entire data against the number of selected features, and as can be seen in Figure 1, L1 regularization excels at reducing feature redundancy. The other methods tend to rather increase the average correlation as fewer features were selected (perhaps because of falling into the trap of selecting highly correlated high-importance features). We expect other greedy-heuristic feature selection methods may behave similarly.

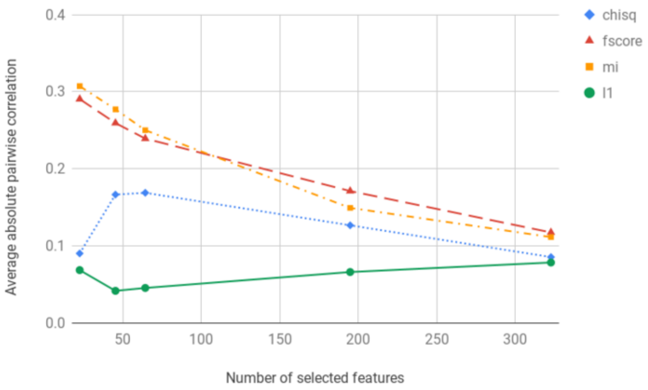


Figure 1: Number of selected features vs. average absolute pairwise correlation of features (ICEWS)

It is also interesting to consider how many features end up being selected by L1. Unlike the other methods, which set this number directly, for L1 it is a property resulting indirectly from the setting of the regularization hyperparameter. This may give L1 the flexibility to adjust the number of features dynamically depending on the magnitude of the drift, while still avoiding feature redundancy.

Table 2: Area under the Learning Curve (ALC) with various feature selection strategies

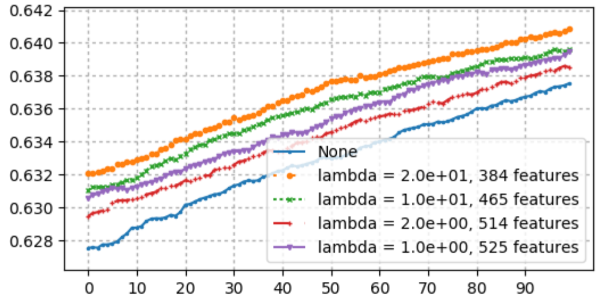
Dataset	#Feats	No Sel.	**L1 (ours)**	ChiSq	ANOVA	MI
`DPC`	188	**33.67**	33.32	32.42	33.27	27.53
	65	33.67	**33.97**	31.33	28.48	21.54
`ERV`	50	42.77	**43.81**	43.34	41.04	24.09
`INS`	88	44.51	**45.03**	42.82	37.85	33.82
`REB`	56	43.89	**45.27**	42.53	33.51	22.63
`IC`	140	43.42	**43.44**	41.77	39.54	36.75

Table 3: Recovery Speed($R97$) of the baseline and our method

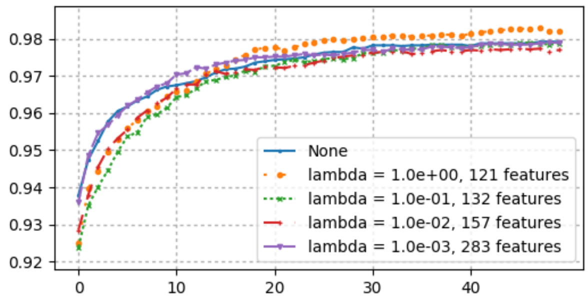
Dataset	#Feats	No Sel.	**L1 (ours)**
`DPC`	188	**23**	38
	65	23	**8**
`ERV`	50	17	**2**
`INS`	88	10	**4**
`REB`	56	25	**3**
`IC`	140	2	**1**

6.3 Results on other data sets

Figure 2 and Table 4 show the results on the airlines and the spam data sets. Both datasets were more sensitive to the hyperparameter than the ICEWS dataset, needing more than just the hand-picked value we used. Once tuned our method could recover 99% of its predictive capability with approximately 65% less labeled instances with the airline dataset. On the other hand our method did not result in any significant improvement for the spam dataset. Unlike the other datasets the selected features did not mitigate the performance loss after drift, resulting in lower initial points as can be seen in Figure 2b. It should also be noted that despite the lower initial performance our method was still able to recover as fast as the baseline (R97, R99 in Table 4) as a result of employing simpler model. A more concrete error analysis remains to be performed in the future.



(a) Airlines



(b) Spam

Figure 2: New labels vs. accuracy recovery after drift in additional datasets

Table 4: ALC and Recovery Speed($R99$) of the baseline and our method in additional datasets

Dataset	#Feats	No Sel. ALC	No Sel. R99	**L1 (ours)** ALC	**L1 (ours)** R99
`airlines`	384	63.29	64	**63.69**	**22**
`spam`	121	48.62	18	**48.65**	**16**
			3(R97)		5(R97)

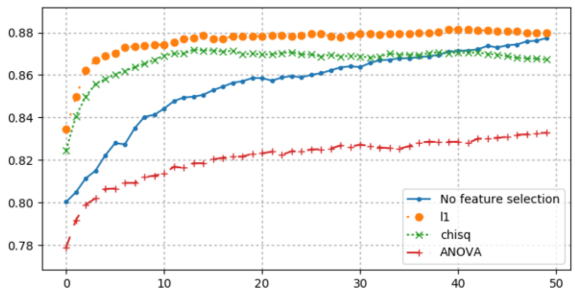
7 CONCLUSIONS AND FUTURE WORK

In the experiments discussed above, we have shown that a judicious feature set reduction, while not paying a large absolute performance penalty, allows for a much faster recovery from drift, in addition to reducing the computational load.

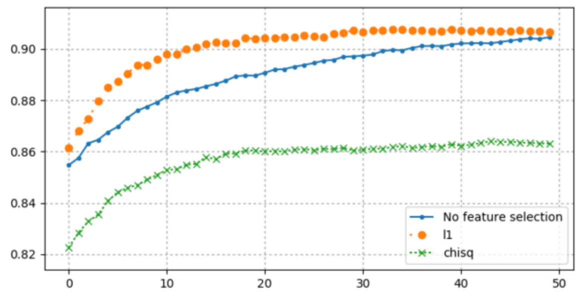
The feature selection method which led to the significantly best recovery results in our experiments was the L1 regularization technique used in our basic sample-driven drift modeling method, as compared to 3 other feature selection methods in the same context.

Future work needs to elucidate how data set properties may impact the effectiveness of L1 (and feature selection in general).

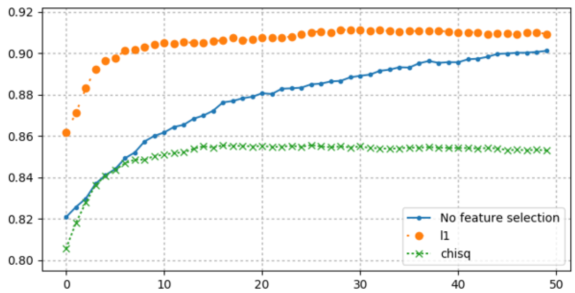
Improvements of our method may use more sophisticated transfer learning methods, combine different approaches, and adapt the feature set dynamically.



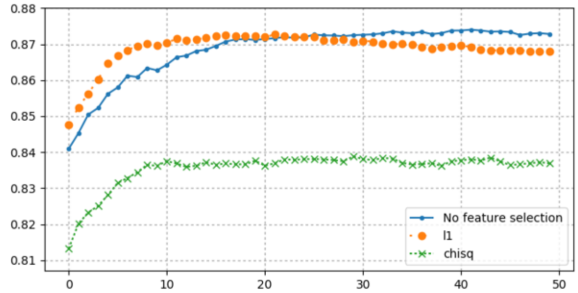
(a) ERV with 50 features



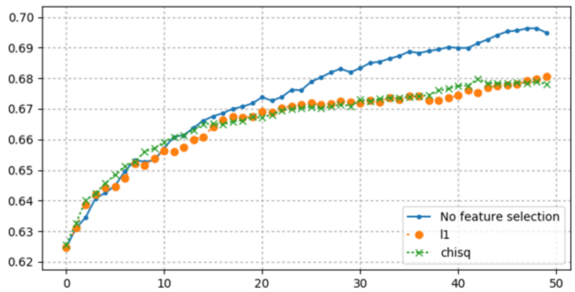
(b) INS with 88 features



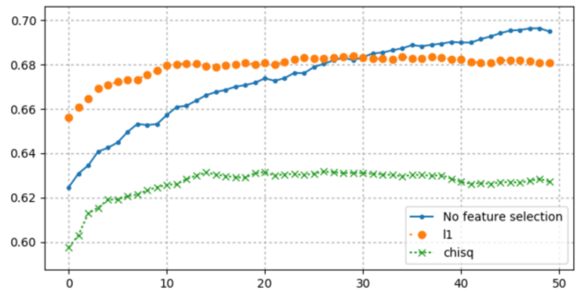
(c) REB with 56 features



(d) IC with 140 features

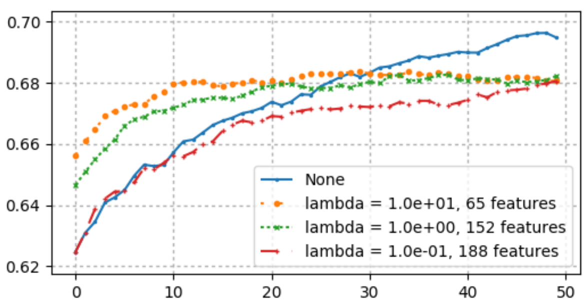


(e) DPC with 188 features

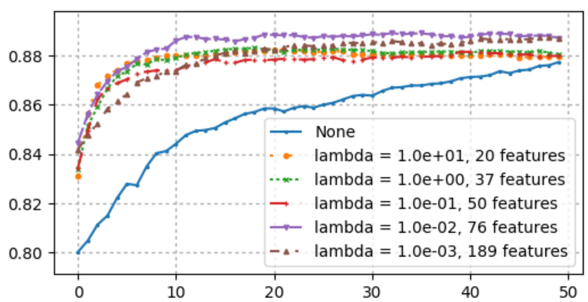


(f) DPC with 65 features

Figure 3: Number of new labeled instances vs. F1 recovery (averaged over 58 epochs) after drift for different classes (ICEWS)



(a) DPC



(b) ERV

Figure 4: Sensitivity analysis of regularization hyperparameters

REFERENCES

[1] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu, 'A framework for clustering evolving data streams', in *VLDB*, (2003).

[2] Jean Paul Barddal, Heitor Murilo Gomes, and Fabrício Enembreck, 'A survey on feature drift adaptation', *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, 1053–1060, (2015).

[3] Jean Paul Barddal, Heitor Murilo Gomes, Fabrício Enembreck, and Bernhard Pfahringer, 'A survey on feature drift adaptation: Definition, benchmark, challenges and future directions', *Journal of Systems and Software*, **127**, 278–294, (2017).

[4] Albert Bifet, Jesse Read, Indrė Žliobaitė, Bernhard Pfahringer, and Geoff Holmes, 'Pitfalls in benchmarking data stream classification and how to avoid them', in *ECML/PKDD*, (2013).

[5] Avrim Blum and Pat Langley, 'Selection of relevant features and examples in machine learning', *Artif. Intell.*, **97**, 245–271, (1997).

[6] José M. Carmona-Cejudo, Gladys Castillo, Manuel Baena-García, and Rafael Morales Bueno, 'A comparative study on feature selection and adaptive strategies for email foldering', *2011 11th International Conference on Intelligent Systems Design and Applications*, 1294–1299, (2011).

[7] Girish Chandrashekar and Ferat Sahin, 'A survey on feature selection methods', *Computers Electrical Engineering*, **40**, 16–28, (2014).

[8] Kylie Chen, Yun Sing Koh, and Patricia Riddle, 'Proactive drift detection: Predicting concept drifts in data streams using probabilistic networks', *2016 International Joint Conference on Neural Networks (IJCNN)*, 780–787, (2016).

[9] DARPA and Sean O'Brien, 'Crisis early warning and decision support: Contemporary approaches and thoughts on future research', *International Studies Review*, **12**(1), 87–104, (March 2010).

[10] Shubhomoy Das, Md Rakibul Islam, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa, 'Active anomaly detection via ensembles: Insights, algorithms, and interpretability', *CoRR*, **abs/1901.08930**, (2019).

[11] Manoranjan Dash and Huan Liu, 'Feature selection for classification', *Intell. Data Anal.*, **1**, 131–156, (1997).

[12] Michal Dereziński and Badri Narayan Bhaskar, 'Anticipating concept drift in online learning', in *Advances in Neural Information Processing Systems*, (2015).

[13] Pedro M. Domingos and Geoff Hulten, 'Mining high-speed data streams', in *KDD*, (2000).

[14] João Gama and Petr Kosina, 'Tracking recurring concepts with metalearners', in *EPIA*, (2009).

[15] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia, 'A survey on concept drift adaptation', *ACM Computing Surveys (CSUR)*, **46**(4), 44, (2014).

[16] Isabelle Guyon and André Elisseeff, 'An introduction to variable and feature selection', *Journal of Machine Learning Research*, **3**, 1157–1182, (2003).

[17] David J. Hand, 'Classifier technology and the illusion of progress', *Statistical Science*, **21**(1), 1–15, (2001).

[18] Jingrui He and Jaime G Carbonell, 'Rare class discovery based on active learning', (2008).

[19] Anil K. Jain and Douglas E. Zongker, 'Feature selection: Evaluation, application, and small sample performance', *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**, 153–158, (1997).

[20] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas, 'Tracking recurring contexts using ensemble classifiers: an application to email filtering', *Knowledge and Information Systems*, **22**(3), 371–391, (2010).

[21] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis P. Vlahavas, 'On the utility of incremental feature selection for the classification of textual data streams', in *Panhellenic Conference on Informatics*, (2005).

[22] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis P. Vlahavas, 'Dynamic feature space and incremental feature selection for the classification of textual data streams', (2006).

[23] Tadeusz J. Kawecki and Dieter Ebert, 'Conceptual issues in local adaptation', (2004).

[24] Daphne Koller and Mehran Sahami, 'Toward optimal feature selection', in *ICML*, (1996).

[25] J. Zico Kolter and Marcus A. Maloof, 'Dynamic weighted majority: A new ensemble method for tracking concept drift', in *ICDM*, (2003).

[26] Petr Kosina and João Gama, 'Very fast decision rules for classification in data streams', *Data Mining and Knowledge Discovery*, **29**, 168–202, (2013).

[27] Petr Kosina, João Gama, and Raquel Sebastião, 'Drift severity metric', in *ECAI*, (2010).

[28] Bartosz Krawczyk and Przemyslaw Skryjomski, 'Cost-sensitive perceptron decision trees for imbalanced drifting data streams', in *ECML/PKDD*, (2017).

[29] Bartosz Kurlej and Michal Wozniak, 'Learning curve in concept drift while using active learning paradigm', in *ICAIS 2011*, (2011).

[30] Pat Langley, 'Selection of relevant features in machine learning', (1994).

[31] Anjin Liu, Yiliao Song, Guangquan Zhang, and Jie Lu, 'Regional concept drift detection and density synchronized drift adaptation', in *IJCAI*, (2017).

[32] Huan Liu, Edward R. Dougherty, Jennifer G. Dy, Kari Torkkola, Eugene Tuv, Hanchuan Peng, Chris H. Q. Ding, Fuhui Long, Michael E. Berens, Lance Parsons, Zheng Zhao, Lei Yu, and George Forman, 'Evolving feature selection', *IEEE Intelligent Systems*, **20**, 64–76, (2005).

[33] Yang Lu, Yiu-Ming Cheung, and Yuan Yan Tang, 'Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift', in *IJCAI*, (2017).

[34] José Ramon Méndez, Florentino Fernández Riverola, Eva Lorenzo Iglesias, Fernando Díaz, and Juan Manuel Corchado, 'Tracking concept drift at feature selection stage in spamhunting: An anti-spam instance-based reasoning system', in *ECCBR*, (2006).

[35] Leandro L. Minku, Nitesh Chawla, and Xin Yao, 'Proceedings of the ijcai 2017 workshop on learning in the presence of class imbalance and concept drift (lpcicd'17)', *CoRR*, **abs/1707.09425**, (2017).

[36] Pierre-Alexandre Murena and Antoine Cornuéjols, 'Minimum description length principle applied to structure adaptation for classification under concept drift', *2016 International Joint Conference on Neural Networks (IJCNN)*, 2842–2849, (2016).

[37] Kajal Naidu, Aparna Dhenge, and Kapil Wankhade, 'Feature selection algorithm for improving the performance of classification: A survey', *2014 Fourth International Conference on Communication Systems and Network Technologies*, 468–471, (2014).

[38] Patrenahalli M. Narendra and Keinosuke Fukunaga, 'A branch and bound algorithm for feature subset selection', *IEEE Transactions on Computers*, **C-26**, 917–922, (1977).

[39] Andrew Y. Ng, 'Feature selection, l1 vs. l2 regularization, and rotational invariance', in *ICML '04*, (2004).

[40] Feiping Nie, Heng Huang, Xiao Cai, and Chris H. Q. Ding, 'Efficient and robust feature selection via joint 2, 1-norms minimization', in *NIPS*, (2010).

[41] Parang Saraf and Naren Ramakrishnan, 'Embers autogsr: Automated coding of civil unrest events', in *KDD*, (2016).

[42] Philip A. Schrodt, 'Automated production of high-volume , near-real-time political event data', (2011).

[43] Le Song, Alexander J. Smola, Arthur Gretton, Justin Bedo, and Karsten M. Borgwardt, 'Feature selection via dependence maximization', *Journal of Machine Learning Research*, **13**, 1393–1434, (2012).

[44] S. Vanaja and K. R. Ramesh Kumar, 'Analysis of feature selection algorithms on classification: A survey', (2014).

[45] Shuo Wang, Leandro L. Minku, and Xin Yao, 'A systematic study of online class imbalance learning with concept drift', *IEEE Transactions on Neural Networks and Learning Systems*, **29**, 4802–4821, (2018).

[46] G.I. Webb, R. Hyde, H. Cao, H.L. Nguyen, and F. Petitjean, 'Characterizing concept drift', *Data Mining and Knowledge Discovery*, (2016).

[47] Daegun Won, Peter J. Jansen, and Jaime G. Carbonell, 'Temporal transfer learning for drift adaptation', in *ESANN*, (2018).

[48] Liu Yang, 'Active learning with a drifting distribution', in *NIPS*, (2011).

[49] Yiming Yang and Jan O. Pedersen, 'A comparative study on feature selection in text categorization', in *ICML*, (1997).

[50] Ying Yang, Xindong Wu, and Xingquan Zhu, 'Mining in anticipation for concept change: Proactive-reactive prediction in data streams', *Data Mining and Knowledge Discovery*, **13**, 261–289, (2006).

[51] Lei Yu and Huan Liu, 'Efficient feature selection via analysis of relevance and redundancy', *Journal of Machine Learning Research*, **5**, 1205–1224, (2004).

[52] Indrė Žliobaitė, Mykola Pechenizkiy, and João Gama, 'An overview of concept drift applications', (2016).