Tracking Concept Drift Using a Constrained Penalized Regression Combiner

Li-Yu Wang^a, Cheolwoo Park^a, Kyupil Yeon^b, Hosik Choi^{c,*}

^a*Department of Statistics, University of Georgia, Athens, GA 30602, USA.*

^b*Department of Applied Statistics, Hoseo University, Asan, Chungnam, 31499, Korea.*

^c*Department of Applied Information Statistics, Kyonggi University, Suwon, Kyonggi-do, 16227, Korea.*

Abstract

The objective of this work is to develop a predictive model when data batches are collected in a sequential manner. With streaming data, information is constantly being updated and a major statistical challenge for these types of data is that the underlying distribution and the true input-output dependency might change over time, a phenomenon known as concept drift. The concept drift phenomenon makes the learning process complicated because a predictive model constructed on the past data is no longer consistent with new examples. In order to effectively track concept drift, we propose model-combining methods using constrained and penalized regression that possesses a grouping property. The new learning methods enable us to select data batches as a group that are relevant to the current one, reduce the effects of irrelevant batches, and adaptively reflect the degree of concept drift emerging in data streams. We demonstrate the finite sample performance of the proposed method using simulated and real examples. The analytical and

*Corresponding author

Email address: `choi.hosik@gmail.com` (Hosik Choi)

empirical results indicate that the proposed methods can effectively adapt to various types of concept drift.

Keywords: Ensemble methods, Clustering, Streaming data

1. Introduction

We consider a sequential regression problem in data streams which constitute a series of data batches entering over time continuously. Each data batch has observations, i.e. input-output pairs, which are to be used to construct a learning model for predicting the future observations. One of the main concerns in the analysis of data streams is the so called *concept drift* phenomenon, in which the target concept to be predicted, e.g. input-output relationship, changes over time. More generally, the underlying distribution that generates data might not be the same for different batches.

The term concept drift has been informally used for referring to any changes of circumstance in a learning process. Kelly et al. (1999) used *population drift* to describe changes in the probability distributions in a classification problem, such as class priors, distributions of classes, and posterior distributions of class memberships. Widmer and Kubat (1996) stated that change of target concept can be induced by changes in the hidden contexts, not given explicitly in the form of predictive features. Lazarescu et al. (2004) defined concept drift in terms of consistency and persistence. Yang et al. (2005) used the term *concept change* and *sampling change*, each of which corresponds to the real concept drift and virtual drift, and divided concept change into concept drift and concept shift to denote gradual drift and abrupt drift, respectively. Tsymbal (2004) stated that from a practi-

cal point of view it is not essential to discriminate real and virtual concept drift since the current model needs to be adjusted in both cases. Kuncheva (2004) summarized the types of change roughly into random noise, gradual changes, abrupt changes, and recurring contexts, depending on which strategy for updating the current model are required. It is noted in the paper that the noise must not be modeled but filtered out, and past knowledge should be reused for stable contexts. It is further noted that a moving window on the training data sequences may be appropriate for a gradual change, but rebuilding a classifier entirely may be preferred for an abrupt change. Therefore, regardless of the degree of drift, the current predictive model should be adjusted because past models will not produce results consistent with the current inputs.

Many researchers have suggested theoretical insights and practical learning algorithms appropriate for concept drift. Bartlett (1992) investigated the learning theory when the probability distribution of the data changes slowly but continuously throughout the learning process. Others (Valiant and Waltz, 1984; Kuh et al., 1990; Helmbold and Long, 1994) have conducted research on the ability to learn drifting concepts under restrictions on the type of admissible concept changes from a viewpoint of PAC (probably approximately correct) learning. From a practical point of view, however, these theoretical results are related only to a gradual concept drift and are driven from a simple, time window-based tracking of the change.

Practical learning algorithms for drifting concepts can be categorized into single predictive model approach and model ensemble approach. In the single model approach, it is essential to select past examples or data batches

consistent with the concept that is reflected in the most recent data batch; thus, an adaptive time window is frequently used to select or weight batches. See Kubat and Widmer (1995), Widmer and Kubat (1996), Klinkenberg and Joachims (2000), Klinkenberg (2004) for examples of the single model approach. In contrast, the ensemble approach combines multiple base models and constructs or restructures an ensemble learner as a final predictive model. Relevant algorithms include Street and Kim (2001), Wang et al. (2003), Kolter and Maloof (2007), and Yeon et al. (2010). Because ensemble methods for concept drift are typically a weighted average of base learners' outputs, the method used to determine weights is critical. The algorithm proposed by Yeon et al. (2010) is to estimate the weights by training a combiner that is a constrained ridge regression applied to the level-1 data set as in stacked generalization (Wolpert, 1992). Generally, ensemble methods produce better prediction accuracy than single model approaches (Yeon et al., 2010), and they work well whether there is no or sharp concept drift.

The objective of the proposed work is to develop an ensemble method that is adaptive to both no or sudden concept drifts. We use a penalized regression combiner for the following reasons. When there are no (or little) drifts, the outputs of base models would be similar across batches, which causes high correlation among them. Thus, the weights corresponding to a similar cluster of batches will be approximately equal during this period. In contrast, if a sudden drift occurs, the outputs right after the change would be completely different from those of other batches. In this case, a combiner should be able to select the batches relevant to the current one and reduce the effects of other irrelevant batches. Yeon et al. (2010) showed that the constrained

ridge regression combiner possesses some of the aforementioned desirable properties in that it can track drifting concepts well. However, their method does not select batches as groups, and the estimated weights do not seem to appropriately represent the current concept changes, as demonstrated in our numerical study. This motivates us to propose a fused penalized regression combiner with constraints that can take both group batch selection and multicollinearity into account. It groups base models with similar contributions to predicting the current target concept by forcing the weights of all the models within a group to be equal. We demonstrate its superior performance over existing methods via simulated and real examples.

The rest of the paper is organized as follows. The next section introduces the general framework of constrained penalized regression combiner for drifting concepts in a regression setting. It also discusses the ridge regression combiner proposed by Yeon et al. (2010) in detail. In Section 3, we propose a new ensemble combiner based on fused penalized regression and illustrate the implementation of its algorithm. We also analytically show the desirable properties of the proposed method. We compare the empirical performance of the proposed method with others using simulated and real examples in Section 4. Finally, we conclude and discuss a possible extension to classification problems in Section 5.

2. Constrained Penalized Regression Approach

We introduce a constrained penalized regression aggregation approach in a regression setting. We borrow notations from Yeon et al. (2010) to show our generalization of their approach. Let $\{D_m, m = 1, 2, \dots, M\}$ be

a sequence of data batches, each consisting of input-output pairs. Suppose that observations in D_m are random samples from unknown distributions $F_m(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{R}^p$ is a predictor vector and $y \in \mathbb{R}$ is a response, i.e. they arise from a statistical model

$$y = f_m(\mathbf{x}) + \epsilon,$$

where ϵ is a random error satisfying $E(\epsilon) = 0$ and is independent of $\mathbf{x}$. The objective of learning is to construct a predictive model for future instances based on the batches $D_1, \dots, D_M$. Note that the underlying distribution of generating data can be different for each batch, which implies the relationship between the response and predictors can change over time.

For prediction we obtain a training set from the outputs of base models and the most recent data batch $D_M = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ that best reflects the target concept to be tracked. For each $\mathbf{x}_i$, $i = 1, \dots, n$, we denote the outputs of base models by $\hat{\mathbf{f}}(\mathbf{x}_i) = (\hat{f}_1(\mathbf{x}_i), \dots, \hat{f}_{M-1}(\mathbf{x}_i), \hat{f}_M^{(-i)}(\mathbf{x}_i))^T$, where $\hat{f}_M^{(-i)}(\mathbf{x}_i)$ is a leave-one-out estimate of $\hat{f}_M(\mathbf{x}_i)$ to avoid an over-fitting problem in D_M . Then, the design matrix and the response for the model aggregation are given as

$$\mathbf{X} = \begin{pmatrix} \hat{f}_1(\mathbf{x}_1) & \cdots & \hat{f}_{M-1}(\mathbf{x}_1) & \hat{f}_M^{(-1)}(\mathbf{x}_1) \\ \hat{f}_1(\mathbf{x}_2) & \cdots & \hat{f}_{M-1}(\mathbf{x}_2) & \hat{f}_M^{(-2)}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{f}_1(\mathbf{x}_n) & \cdots & \hat{f}_{M-1}(\mathbf{x}_n) & \hat{f}_M^{(-n)}(\mathbf{x}_n) \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad (2.1)$$

and the final predictive model has the form of $f(\mathbf{x}) = \sum_{m=1}^M w_m \hat{f}_m(\mathbf{x})$ where $\mathbf{w} = (w_1, \dots, w_M)^T$ is the weight vector. Hence, the objective of a model ensemble approach is to optimally determine the weight values $(w_1, \dots, w_M)$

using $\hat{\mathbf{f}}(\mathbf{x}_i)$. We note that, in the case of no concept drift, the columns of the design matrix in (2.1) will be correlated due to similar outputs across batches. On the other hand, in the case of abrupt change or recurring events, some columns in the design matrix will be irrelevant, and thus selection of columns (i.e. batches) would be desirable.

The regression approach proposed by Chu et al. (2004) applied the least squares estimation approach to determine $\mathbf{w}$ because it can track sudden changes effectively. However, it does not provide the optimal weights in the case of no concept drift and potentially negative weight values can cause an interpretation problem. A weighted average that guarantees non-negative weights, e.g. Wang et al. (2003), can perform well particularly when there exists no (or little) concept drift. However, it has been shown, and also is demonstrated in Section 4, that the prediction accuracy right after a sudden change can be poor and its recovery is also slow. To simultaneously deal with the multicollinearity and selection problems, Yeon et al. (2010) proposed the constrained ridge regression aggregation method that combines the desirable properties of the regression-type and weighted average-type ensemble methods. While this method can cope with various types of concept drift in terms of prediction accuracy, the estimated weights do not properly measure the contribution of each base model. In order to improve their method, we consider a new class of constrained penalized regression combiners that includes the existing methods as special cases.

In order to determine $\mathbf{w}$, we propose to use a constrained penalized aggregation method that minimizes the following objective function with respect

to w_m for $m = 1, \dots, M$:

$$R_\gamma^{\mathbf{D}}(\mathbf{w}) = \sum_{i=1}^n \left(y_i - \sum_{m=1}^M w_m \hat{f}_{im} \right)^2 + \lambda \|\mathbf{D}\mathbf{w}\|_\gamma^\gamma, \quad (2.2)$$

subject to $\sum_{m=1}^M w_m = 1, w_m \geq 0,$

where $\|\mathbf{w}\|_\gamma = (\sum_{m=1}^M w_m^\gamma)^{1/\gamma}$ with $\gamma > 0$, $\mathbf{D}$ is a $M \times M$ matrix, λ is a tuning parameter, and $\hat{f}_{im}$ is the m th element of the vector $\hat{\mathbf{f}}(\mathbf{x}_i)$. If $\lambda = 0$, it corresponds to a weighted average type combiner, and if $\lambda = 0$ with no constraint on $\mathbf{w}$, it corresponds to the least squares estimation regression combiner (Chu et al., 2004). Also, if $\gamma = 2$ and $\mathbf{D} = \mathbf{I}$, the $M \times M$ identity matrix, then $R_\gamma^{\mathbf{D}}(\mathbf{w})$ in (2.2) becomes (Yeon et al., 2010)

$$R_2^{\mathbf{I}}(\mathbf{w}) = \sum_{i=1}^n \left(y_i - \sum_{m=1}^M w_m \hat{f}_{im} \right)^2 + \lambda \sum_{m=1}^M w_m^2, \quad (2.3)$$

subject to $\sum_{m=1}^M w_m = 1, w_m \geq 0.$

Yeon et al. (2010) showed that this combiner with l_2 penalty is able to effectively track concept drifts using the measure of angle. In the no concept drift case each base model is expected to make a similar contribution to the final model combiner, and thus the optimal aggregation is the simple average of base models, i.e. $\mathbf{w}^* = (1/M, \dots, 1/M)^T$. On the other hand, if a considerable concept drift occurs, the contributions before and after the change can be significantly different, and thus the estimated weight $\hat{\mathbf{w}}$ should be far from $\mathbf{w}^*$. The degree of departure of $\hat{\mathbf{w}}$ from $\mathbf{w}^*$ represents the degree of concept drift, and this can be captured through the angle between the two vectors:

Definition 1. (A measure of concept drift, Yeon et al. (2010)) For $\mathbf{w}^* = (1/M, \dots, 1/M)^T$ and $\hat{\mathbf{w}}$ obtained by a combiner, we define the degree of concept drift as the angle between the two vectors given by

$$\eta(\mathbf{w}^*, \hat{\mathbf{w}}) = \cos^{-1} \left(\frac{|\langle \mathbf{w}^*, \hat{\mathbf{w}} \rangle|}{\|\mathbf{w}^*\|_2 \|\hat{\mathbf{w}}\|_2} \right) \quad (2.4)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors in $\mathbb{R}^M$.

Using this definition, Yeon et al. (2010) showed that the ridge regression combiner obtained from (2.3) possesses the following good properties; (i) For the no concept drift case, the angle $\eta(\mathbf{w}^*, \hat{\mathbf{w}})$ is small when λ is sufficiently large, and thus the optimal weight $\mathbf{w}^*$ can be obtained. (ii) λ controls the adaptivity of the algorithm to concept drift, and it provides a weight vector $\hat{\mathbf{w}}$ that makes the angle $\eta(\mathbf{w}^*, \hat{\mathbf{w}})$ large if a considerable concept drift occurs. However, the method leaves room for improvement. One drawback of the ridge regression combiner is lack of the grouping property when a subset of the past batches contribute equally to future observations. When there is no concept drift for a certain period of time, the corresponding batches could be viewed as part of a single cluster, each with approximately equal weight. For instance, if a sudden concept drift is observed at the m th batch and the data are stabilized afterwards, $\mathbf{w} = (0, \dots, 0, 1/(M - m), \dots, 1/(M - m))^T$ would provide optimal solution. In such a case, two clusters seem to exist whose corresponding weights are 0 and $1/(M - m)$, respectively. In a regression setting, the idea of clustering predictors has been studied; Tibshirani et al. (2005) proposed the fused lasso, which attempts to group neighboring predictors by either time or location. Park et al. (2007) also proposed a two-step procedure that creates new predictors by averaging the

cluster obtained by hierarchical clustering analysis. In the next section, we propose ensemble methods that are able to identify all possible homogeneous batches.

3. Generalized Fused Lasso

In this section, we propose two model combining methods with l_1 and l_2 penalties that work with the concept drift measure defined in Definition 1 and possess a data-driven grouping property.

Let us first consider the following pairwise difference l_2 penalty:

$$P(\mathbf{w}) = \sum_{l < m} (w_l - w_m)^2 = \mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w},$$

where $\mathbf{D}$ is a penalty matrix having l th row of $\mathbf{D}$, $\mathbf{d}_l = (0, \dots, 1, \dots, -1, \dots, 0)^T \in \mathbb{R}^M$ for $l = 1, \dots, M'$ where $M' = \binom{M}{2}$, i.e., $\mathbf{d}_l$ is a row vector containing all zeros except for 1 and -1 in the i th and j th components, respectively. For example $M = 3$,

$$\mathbf{D} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}.$$

Therefore, the objective function is given as

$$\begin{aligned} R_2^{\mathbf{D}}(\mathbf{w}) &= \sum_{i=1}^n \left(y_i - \sum_{m=1}^M w_m \hat{f}_{im} \right)^2 + \lambda \sum_{l < m} (w_l - w_m)^2, \quad (3.1) \\ \text{subject to } &\sum_{m=1}^M w_m = 1, w_m \geq 0. \end{aligned}$$

One can consider more general difference penalty, $P(\mathbf{w}) = \mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w}$ with $\mathbf{D} \mathbf{1}_M = 0$. We focus on the pairwise difference penalty in this paper.

Intuitively, it forces the method to have the grouping property by penalizing a large difference between two weights. In the following two theorems, we show that the proposed method also has the desirable properties as in Yeon et al. (2010); the proposed method with the pairwise difference l_2 penalty produces the equal weight in the case of no concept drift and the degree of concept drift is reflected by a weight vector obtained from (3.1) and is controlled by λ . The proofs to Theorems 1 and 2 are provided in the Appendix.

Theorem 1. *When concept drift is not present, the minimizer of (3.1) is given as $\hat{\mathbf{w}} = (1/M, \dots, 1/M)^T$ for sufficiently large λ .*

Theorem 2. *For $\hat{\mathbf{w}}$ obtained from (3.1), $\eta(\mathbf{w}^*, \hat{\mathbf{w}})$ in Definition 1 is monotonically decreasing as λ increases.*

The proposed method also possesses the sparsity property that enables us to select relevant batches to the current one.

Proposition 1. *For $M = 2$, the optimal ensemble minimizing*

$$\sum_{i=1}^n (y_i - w_1 \hat{f}_{i1} - w_2 \hat{f}_{i2})^2 + \lambda(w_1 - w_2)^2, w_1 + w_2 = 1, w_1 \geq 0, w_2 \geq 0$$

is given as

$$w_1 = \begin{cases} z & \text{if } -2\lambda < r < v + 2\lambda \\ 0 & \text{if } r \leq -2\lambda \\ 1 & \text{if } r \geq v + 2\lambda \end{cases}$$

where $r = \sum_{i=1}^n (\hat{f}_{i1} - \hat{f}_{i2})(y_i - \hat{f}_{i2})$, $v = \sum_{i=1}^n (\hat{f}_{i1} - \hat{f}_{i2})^2$, and $z = (r + 2\lambda)/(v + 4\lambda)$.

This result implies that the constraint on the weight produces the sparse solution even for the l_2 penalty. When r value is less than -2λ , the optimal solution is $(w_1^*, w_2^*) = (0, 1)$, and if $r \geq v + 2\lambda$, the optimal solution is $(w_1^*, w_2^*) = (1, 0)$. Also, the optimal solution is equally weighted for sufficiently large λ . In other words, if we let $w_1(\lambda)$ be the solution at given λ for sufficiently large λ , $w_1(\lambda) = 1/2$. Intuitively, if the angle between two vectors, $\mathbf{y} - \mathbf{f}_2$ and $\mathbf{f}_1 - \mathbf{f}_2$, is in the range of 90° and 270° , then r is negative. In this case, the $\mathbf{f}_2$ vector is much closer to $\mathbf{y}$, and thus the optimal solution of w_1 is zero. This property is useful because the proposed method reduces the impact of irrelevant batches of data when it aggregates the information, which is critical particularly when concept changes abruptly.

In what follows we further consider the optimization problem with $\gamma = 1$ in (2.2). The l_1 penalty is more appropriate for batch clustering than the l_2 penalty, as demonstrated in Section 4. Recently, Kim (2009) proposed l_1 regularized trend filtering and Arnold and Tibshirani (2014) developed the dual path algorithm that allows fast computation of the general lasso problem including trend filtering problem.

The proposed method with l_1 penalty is to minimize

$$R_1^D(\mathbf{w}) = \sum_{i=1}^n \left(y_i - \sum_{m=1}^M w_m \hat{f}_{im} \right)^2 + \lambda \|\mathbf{D}\mathbf{w}\|_1, \quad (3.2)$$

subject to $\sum_{m=1}^M w_m = 1, w_m \geq 0.$

It is referred to as “MC.Lasso1+” indicating *Model Combining* via general *Lasso* regression with constraints of sum-to-1 and non-negativity (+). Theorem 3 indicates that the proposed method also can generate a simple average

model in the case of no concept drift for a sufficiently large λ . The proof of Theorem 3 is shown in the Appendix.

Theorem 3. *The minimizer $\hat{w}_m$ from (3.2), $m = 1, \dots, M$, is given as $1/M$ for sufficiently large λ .*

3.1. Computational Algorithm

In this subsection, we present an optimization algorithm to minimize the objective function in (2.2) when $\gamma = 1$ and 2. Let $l(\mathbf{w})$ be the convex loss function, and $\mathcal{C} = \{w_l : \sum_l w_l = 1, w_l \geq 0\}$ be the positive convex cone. When $l(\mathbf{w})$ is the squared loss function with no constraint, the objective function can be efficiently solved by recently developed methods including entire solution path algorithms (e.g., Tibshirani and Taylor (2011), Shen and Huang (2010) etc). However, we encounter a technical computational difficulty due to the positive cone constraint $\mathbf{w} \in \mathcal{C}$.

We resolve this difficulty by the proposed optimization algorithm which consists of two routines: outer routine and inner routine. The outer routine adopts the alternating direction method of multipliers (ADMM) (Boyd et al., 2010) algorithm, and the inner routine consists of proximal gradient methods with some positive cone constraint. Since the lasso-type penalty term is not differentiable, there is no closed-form solution. The ADMM enables us to split the problem into two subproblems which are much easier to solve than the original problem. We note that the problem (2.2)

$$\min_{\mathbf{w} \in \mathcal{C}} l(\mathbf{w}) + \lambda \|\mathbf{D}\mathbf{w}\|_\gamma^\gamma$$

is equivalent to the following problem

$$\min_{\mathbf{w} \in \mathcal{C}, \mathbf{e}} l(\mathbf{w}) + \lambda \|\mathbf{e}\|_\gamma^\gamma, \text{ subject to } \mathbf{e} = \mathbf{D}\mathbf{w}, \quad (3.3)$$

where $\mathbf{e} = (e_1, e_2, \dots, e_{M'})^T$ with $M' = \binom{M}{2}$. An auxiliary variable e_l , which is equal to the penalty $\mathbf{d}_l^T \mathbf{w}$, has the range $-1 \leq e_l \leq 1$, for all l since the pairwise difference of w_m and w_l lies on $[-1, 1]$. We can then rewrite the optimization problem in (3.3) as the following augmented Lagrangian function with parameters $\mathbf{w}$ and $\mathbf{e} = (e_1, e_2, \dots, e_{M'})^T$: for some positive ρ ,

$$l(\mathbf{w}) + \lambda \|\mathbf{e}\|_\gamma^\gamma + \boldsymbol{\delta}^T (\mathbf{e} - \mathbf{D}\mathbf{w}) + \frac{\rho}{2} \|\mathbf{e} - \mathbf{D}\mathbf{w}\|_2^2 \quad (3.4)$$

where $\boldsymbol{\delta}$ is the Lagrangian parameter and ρ is the augmented penalty. The updating procedure through ADMM (Boyd et al., 2010) to solve the problem (3.4) is as follows. For $t = 0, 1, \dots$,

1. Primal update: $\mathbf{w}^{(t+1)} = \underset{\mathbf{w} \in \mathcal{C}}{\operatorname{argmin}} l(\mathbf{w}) + \frac{\rho}{2} \|(\mathbf{e}^{(t)} + \rho^{-1} \boldsymbol{\delta}^{(t)}) - \mathbf{D}\mathbf{w}\|_2^2$.
2. Dual update: for all l ,

$$\mathbf{e}^{(t+1)} = \underset{\mathbf{e}}{\operatorname{argmin}} \lambda \|\mathbf{e}\|_\gamma^\gamma + \frac{\rho}{2} \|\mathbf{e} - (\mathbf{D}\mathbf{w}^{(t+1)} - \rho^{-1} \boldsymbol{\delta}^{(t)})\|_2^2, \text{ subject to } |e_l| \leq 1.$$
3. Lagrangian multiplier update: $\boldsymbol{\delta}^{(t+1)} = \boldsymbol{\delta}^{(t)} + \rho(\mathbf{e}^{(t+1)} - \mathbf{D}\mathbf{w}^{(t+1)})$.

The problem in dual update is additive elementwise along with e_l . That is,

$$\begin{aligned} & \min_{\mathbf{e}} \lambda \|\mathbf{e}\|_\gamma^\gamma + \frac{\rho}{2} \|\mathbf{e} - (\mathbf{D}\mathbf{w}^{(t+1)} - \rho^{-1} \boldsymbol{\delta}^{(t)})\|_2^2, \text{ subject to } |e_l| \leq 1, \text{ for all } l \\ &= \sum_{\{e_l: |e_l| \leq 1\}} \min \left(\lambda |e_l|^\gamma + \frac{\rho}{2} (e_l - z_l)^2 \right), \end{aligned}$$

where $\mathbf{z} = \mathbf{D}\mathbf{w}^{(t+1)} - \rho^{-1} \boldsymbol{\delta}^{(t)} = (z_1, z_2, \dots, z_{M'})^T$. This can be solved by quadratic programming or by one dimensional optimization analytically. When $\gamma = 1$, we can derive the following analytical closed-form

$$e_l = \begin{cases} \operatorname{sign}(z_l) \max\{|z_l| - \frac{\lambda}{\rho}, 0\}, & \text{if } |z_l| < (\lambda + \rho)/\rho \\ \operatorname{sign}(z_l), & \text{if } |z_l| \geq (\lambda + \rho)/\rho \end{cases},$$

and when $\gamma = 2$,

$$e_l = \begin{cases} \frac{\rho}{2\lambda + \rho} |z_l|, & \text{if } |z_l| < (2\lambda + \rho)/\rho \\ \text{sign}(z_l), & \text{if } |z_l| \geq (2\lambda + \rho)/\rho \end{cases}.$$

In order to solve Primal update, we apply the proximal gradient method (Parikh and Boyd, 2013). Let

$$f(\mathbf{w}) = l(\mathbf{w}) + \frac{\rho}{2} \|(\mathbf{e}^{(t)} + \rho^{-1} \boldsymbol{\delta}^{(t)}) - \mathbf{D}\mathbf{w}\|_2^2$$

and $\nabla f(\mathbf{w}) = \nabla l(\mathbf{w}) + \rho \mathbf{D}^T (\mathbf{D}\mathbf{w} - (\mathbf{e}^{(t)} + \rho^{-1} \boldsymbol{\delta}^{(t)}))$ denotes its gradient at $\mathbf{w}$. Furthermore, $\nabla^2 f(\mathbf{w})$ denotes the second derivative hessian matrix and $f_{L,\mathbf{u}}(\mathbf{w})$ is the linearization of $f(\mathbf{w})$ at $\mathbf{u}$ defined as follows:

$$\begin{aligned} f_{L,\mathbf{u}}(\mathbf{w}) &= f(\mathbf{u}) + \nabla f(\mathbf{u})^T (\mathbf{w} - \mathbf{u}) + \frac{L}{2} \|\mathbf{w} - \mathbf{u}\|_2^2 \\ &\propto f(\mathbf{u}) + \frac{L}{2} \|\mathbf{w} - (\mathbf{u} - L^{-1} \nabla f(\mathbf{u}))\|_2^2. \end{aligned}$$

Therefore, $f(\mathbf{w}) \leq f_{L,\mathbf{u}}(\mathbf{w})$ holds when L is the maximum eigenvalue of the second derivative matrix $\nabla^2 f(\mathbf{w}) = \nabla^2 l(\mathbf{w}) + \rho \mathbf{D}^T \mathbf{D}$.

Given the current solution $\mathbf{u}$, we minimize $f_{L,\mathbf{u}}(\mathbf{w})$ instead of $f(\mathbf{w})$ and iterate the minimization process until convergence. The minimization problem $f_{L,\mathbf{u}}(\mathbf{w})$ is also translated into the proximal gradient problem:

$$\min_{\mathbf{w}} \|\mathbf{w} - \mathbf{v}\|_2^2, \text{ subject to } w_m \geq 0, \text{ for all } m, \sum_{m=1}^M w_m = 1, \quad (3.5)$$

where $\mathbf{v} = \mathbf{u} - L^{-1} \nabla f(\mathbf{u})$.

In the optimization problem (3.5), it is solved efficiently by the algorithm developed by Shalev-Shwartz (2006). The Lagrangian of the optimization

problem (3.5) can be rewritten as

$$Q = \|\mathbf{w} - \mathbf{v}\|_2^2 + \beta \left(\sum_{m=1}^M w_m - 1 \right) - \sum_{m=1}^M \xi_m w_m, \quad (3.6)$$

where $\beta \in \mathbb{R}$ is a Lagrangian multiplier and $\xi_m \in \mathbb{R}_+$ for $m = 1, \dots, M$ are non-negative Lagrangian multipliers. Differentiating equation (3.6) with respect to w_m and setting it to zero, we obtain the optimality condition so called Lagrangian stationarity:

$$\frac{dQ}{dw_m} = 2w_m - 2v_m + \beta - \xi_m = 0.$$

The complementary slackness Karush-Kuhn-Tucker (KKT) condition is

$$\xi_m w_m = 0, m = 1, \dots, M,$$

and this condition implies that whenever $w_m > 0$, we should have $\xi_m = 0$. Thus, if $w_m > 0$ we find that $w_m = v_m - \beta/2$ from the Lagrangian stationarity condition. Here, β is an unknown positive value and some of w_m are positive. We apply the following Lemma 1.

Lemma 1. *(Shalev-Shwartz, 2006) Let $\mathbf{w}$ be the optimal solution to the minimization problem in (3.5). Let a and b be two indices such that $v_a > v_b$. If $w_a = 0$, then w_b must be zero as well.*

Let I denote the set $\{m : w_m > 0\}$ and its cardinality be s . If we know the index set I in advance, then by Lemma 1, β is given as

$$\beta = \frac{2}{s} \left(\sum_{l=1}^s v_{(l)} - 1 \right),$$

where $v_{(l)}$ are ordered values such that $v_{(1)} \geq v_{(2)} \geq \cdots \geq v_{(M)}$. Then, for a given β value, the optimal solution is given as

$$w_{(m)} = \begin{cases} v_{(m)} - \frac{1}{s} \left(\sum_{l=1}^s v_{(l)} - 1 \right) & m \leq s, \\ 0 & m > s. \end{cases} \quad (3.7)$$

Finally, s is given as

$$\max \left\{ m : v_{(m)} - \frac{1}{m} \left(\sum_{r=1}^m v_{(r)} - 1 \right) > 0 \right\}.$$

4. A Numerical Study

This section studies the performance of the proposed method using four simulated examples with various forms of concept drift. The simulation settings are similarly designed to those in Yeon et al. (2010). We compare the proposed method (MC.Lasso1+) in terms of root mean square errors (RMSE) and adaptiveness to concept drift (measured by the angle defined in Definition 1) with three existing methods: MC.WAvg (weighted average method proposed by Wang et al. (2003)), MC.Lse1+ (least squares regression method proposed by Chu et al. (2004) with constraints), MC.Ridge1+ (ridge regression method proposed by Yeon et al. (2010)). In Wang et al. (2003), the weights are determined by the MSE criterion in a classification setting, and thus we modify the MSE as the prediction error in a regression setting. Because Chu et al. (2004) originally used the likelihood of logistic regression in a classification setting, we also modify the likelihood as that of ordinary regression. We do not present the results of the proposed method with the l_2 penalty because we find that it works similarly to MC.Ridge1+ in all examples. In Section 4.2, we present the comparison of these four methods using

two real examples. We use a linear model as a base model for both simulated and real examples.

4.1. Simulated Examples

4.1.1. Setting 1 - Single Abrupt Change Point

We construct 20 training sequential data batches $D_1, \dots, D_{20}$. Each D_m contains 200 observations generated independently from $y = \sum_{i=1}^{10} \alpha_i X_i + \epsilon$ where y is a response variable and $(X_1, \dots, X_{10})$ are independent predictors from the Uniform(0,1) distribution. The noise variable ϵ is generated independently from the standard normal distribution. The concept drift is incorporated into the model by changing the value of $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{10})^T$ starting from a certain time point T . For example, if $T = 10$, each D_m for $m = 1, \dots, 9$ is generated with $\boldsymbol{\alpha} = (5, 5, 5, 5, 5, -2, -2, -2, -2, -2)^T$ and the remaining data batches are with $\boldsymbol{\alpha} = (-2, -2, -2, -2, -2, 5, 5, 5, 5, 5)^T$. We consider ten change points, $T = 0, 3, 5, 7, 9, 11, 13, 15, 17, 19$. We also construct a test data batch which has the same data generation process as the last training data batch D_{20} except for sample size. The test data batch contains $n = 1000$ independent observations. The tuning parameter λ is selected by 10-fold cross-validation in MC.Ridge1+ and MC.Lasso1+ because we find that the results with leave-one-out, 5-fold, and 10-fold cross-validation are similar, and thus want to save computation time. We select λ from a sequence of values, $\log_{10}(0.5 \times \lambda) = c$ where c varies from -3 to 3 by 0.2 . We select the best λ which produces the lowest RMSE. We set the ensemble size $M = 20$, which is equal to the number of data batches by default, but we will study the effect of different sizes later.

Figure 1 shows the angle between the estimated weights and $\mathbf{w}^* = (1/20,$

(a) Angle

(b) RMSE

Figure 1: The left panel (a) shows the angles for MC.Lasso1+ and MC.Ridge1+ at each position of concept drift and the right panel (b) shows the RMSE for four methods at each position of concept drift.

$\dots, 1/20)^T$ for MC.Lasso1+ and MC.Ridge1+ in (a) and the RMSE for the four methods in (b) when there exists one position of concept drift. For MC.Lasso1+, the angle is close to 0 in the case of no concept drift (corresponding to the change point position at 0) and becomes larger when the change point gets closer to the prediction time point. It implies that the estimated weights are close to $\mathbf{w}^*$ at the beginning and moves farther away from $\mathbf{w}^*$ to account for concept drift. MC.Ridge1+ shows a similar pattern with some deviation, but the angle is not close 0 in the case of no concept drift. We will come back to this issue when we discuss their estimated weights. Regarding the prediction accuracy, all three regression-based methods perform similarly, while MC.Ridge1+ and MC.Lasso1+ have smaller RMSEs than MC.Lse1+. Although the RMSE for MC.WAvg is small at the beginning in the case of no or a little concept drift effect, it increases exponentially when the change point gets closer to the current prediction time point.

Figure 2 depicts the estimated weights for four methods at each position of concept drift. It can be seen that MC.Lasso1+ produces equal weights $\mathbf{w}^*$ for all batches when there is no concept drift. When an abrupt change occurs at T , it successfully assigns the zero weight to the base models (or batches) up to $T-1$ and the equal weight to the remaining ones. For example, when $T = 19$, MC.Lasso1+ produces $\hat{\mathbf{w}} = (0, \dots, 0, 0.5, 0.5)^T$, which implies that it only uses the last two base models (or batches) for prediction; this is appropriate because the relationship between the response and predictors in the first 18 batches are different from the test data batch after the abrupt change. MC.WAvg also possesses this property, but it does not produce exactly zero weight for irrelevant batches. MC.Ridge1+ and MC.Lse1+ tend

Figure 2: Estimated weights at each change point for four methods.

to produce a similar pattern, but their weights cannot be interpreted in a clear way. In theory, MC.Ridge1+ is supposed to return equal weights as $\lambda \rightarrow \infty$. However, in practice, the weights are not exactly equal, as demonstrated in Figure 2, because λ is chosen based on the prediction error. This is our motivation of using the fused penalty, because it forces the weights to be equal when they are similar. Hence, MC.Lasso1+ provides a better interpretation of estimated weights as well as a similar or better prediction performance compared to MC.Ridge1+. In summary, MC.Lasso1+ provides a clear interpretation on the estimated weights, possesses a grouping property that clearly separates relevant data batches from the others, and produces accurate prediction for this simulation setting. Overall, it adapts to a single concept drift effectively.

An ensemble size M plays an important role in the ensemble algorithms. Small ensemble sizes focus only on the most recent data batches, whereas larger ensemble sizes will account for more of the data batches. For example, with ensemble size $M = 10$, the weights are computed based on the most recent 10 data batches only. We study the effect of the ensemble size in the following way: we generate 30 data batches under the similar simulation scheme except that we fix the position of the abrupt change at $T = 14$. We consider two ensemble sizes, 10 and 25. Figure 3 shows the batch-by-batch comparisons of the RMSE for the ensemble size 10 and 25. It is evident from Figure 3(a) that the recovery rate after the change at $T = 14$ for MC.WAvg is much slower than the other three regression-based combiners. Among the penalized regression combiners, MC.Lasso1+ and MC.Ridge1+ perform better than MC.Lse1+. When we increase the ensemble size (Figure 3(b)), we see a

Figure 3: The left panel (a) shows the RMSE with ensemble size 10 and the right panel (b) shows the RMSE with ensemble size 25 for four methods when there exists a single abrupt change.

similar result. MC.WAvg and MC.Lse1+ have a slower recovery rate because the larger ensemble size causes them to include batches occurring before the change at $T = 14$. However, the prediction pattern of both MC.Lasso1+ and MC.Ridge1+ stay unchanged for both ensemble sizes. Therefore, it can be concluded that the penalized regression combiners are robust to different ensemble sizes.

Figure 4: The estimated weight plots for batch-to-batch comparison when ensemble size is 10 in the case of a single abrupt change.

Figure 4 presents the estimated weights of four methods with ensemble size $M = 10$. Each panel displays 30 curves because there are originally 30 data batches and the estimated weights are computed with ensemble size $M = 10$ at each data batch. For example, for MC.Lasso1+, $\hat{w}_1 = 1$ at Batch 1 (B_1 in the plot) because only one data batch is available, and $\hat{w}_1 = \hat{w}_2 = 0.5$ at B_2 because there are two data batches and the abrupt change has not yet occurred. Likewise, note that all the data batches are given equal weight (i.e. $\hat{w}_1 = \dots = \hat{w}_{10} = 0.1$) before the abrupt change occurs. After the change at $T = 14$, however, batches before the change are given zero weight and only batches after the change are selected by the regression combiner (e.g. $\hat{w}_1 = \dots = \hat{w}_9 = 0$ and $\hat{w}_{10} = 1$ for B_{15}). From B_{16} , it can be seen that MC.Lasso1+ assigns the zero weight to the irrelevant batches before the change and the equal weight to the relevant batches after the change. MC.WAvg behaves similarly to MC.Lasso1+, but does not perform as effectively as MC.Lasso1+. The other two methods also roughly follow this pattern, but their weights are not clearly interpretable. One can draw a similar conclusion with ensemble size $M = 25$. This again demonstrates the effective adaptiveness of MC.Lasso1+ to a single abrupt change.

Remark 1

- (i) In order to test the robustness of the proposed methods to the independent normality assumption, we tried another simulation setting: Setting 1 with time series models AR(1) and AR(2). Their results were similar to Figures 1 and 2. We do not include them to save space.
- (ii) One can choose a different learner such as regression tree or penalized regression model, and we leave that choice to practitioners depending

on data structure in hand. As demonstrated in Yeon et al. (2010) with MC.Ridge1+, a constrained penalized regression approach provides a better prediction performance than other methods regardless of the choice of learner.

4.1.2. Setting 2 - Two Abrupt Change Points

This simulation is designed to demonstrate adaptiveness of the proposed algorithm in the presence of multiple change points. The data generation process is similar to that of Setting 1 except for the change points and α .

The regression coefficient vector α may change at two time points T between two coefficient sets, $\alpha_1 = (5, 5, 5, 5, 5, -5, -5, -5, -5, -5)^T$ and $\alpha_2 = (-5, -5, -5, -5, -5, 5, 5, 5, 5, 5)^T$. For example, if $T = (2, 6)$, D_m for $m = 1, 6, 7, \dots, 21$ (test batch) are generated with α_1 and D_m for $m = 2, \dots, 5$ are with α_2 . We consider 6 pairs of change points $T = (2, 6), (2, 8), (2, 10), (2, 12), (2, 14),$ and $(2, 16)$. We measure the angle between the $\hat{\mathbf{w}}$ and $\mathbf{w}^* = (1/20, \dots, 1/20)^T$ and RMSEs for four methods.

The results for the angle and RMSE are similar to those of the single change point in Setting 1. Figure 5(a) shows that the angle becomes bigger when the closest change point is near the prediction time point. In Figure 5(b), MC.Lasso1+ performs best in terms of prediction accuracy. MC.WAvg performs competitively when the closest change is not near the prediction time point, but performs worst as the closest changes gets near to the prediction time point.

The weight plots in Figure 6 also show similar results as in Setting 1. MC.Lasso1+ clusters the batches into two groups by yielding the zero weight for the irrelevant batches and the equal weight for the relevant batches.

Figure 5: The left panel (a) shows the angles and the right panel (b) shows the RMSE for four methods at each pair of concept drift positions.

MC.WAvg possesses similar properties, but the estimated weights are slightly different. MC.Ridge1+ and MC.Lse1+ are similar as well, but they are not as clearly interpretable as MC.Lasso1+.

We investigate the effect of the ensemble size in the presence of two abrupt changes. Figure 7 displays the batch-by-batch comparisons of the RMSE with the ensemble size 10 and 25 when 30 data batches are generated. The abrupt change points are set at $T = 3$ and 15. Similar to Setting 1, MC.Lasso1+ and MC.Ridge1+ are robust to different ensemble sizes, but the prediction accuracy of MC.WAvg recovers slowly right after the abrupt change, particularly for $M = 25$.

The weight plots in Figure 8 demonstrate the grouping property in the weights for MC.Lasso1+. It assigns equal weights to batches similar to the current concept and the zero weight to the batches that differ from the current

Figure 6: Estimated weights of four methods for two change points.

Figure 7: The left panel shows the RMSE with the ensemble size 10 and the right panel with 25 for four methods when there exist two abrupt changes.

concept. For example at B_4 , we combine the first four sets of predictions to make prediction for data batch 5. In this case, $\hat{w}_1 = \hat{w}_2 = \hat{w}_3 = 0$ and $\hat{w}_4 = 1$ because the concept changes after the first three data batches. The weight plots for $M = 25$ are not included to save space, but they show a similar pattern.

Remark 2 In Figures 1(b) and 5(b), the performances of MC.Lasso1+ and MC.Ridge1+ look similar in terms of RMSE. We redraw these two plots in Figure 9, including only MC.Lasso1+ and MC.Ridge1+ only for a better comparison of RMSE. In both single and two change points, MC.Lasso1+ consistently produces smaller RMSE values, although the differences are not large. Even when they have the same prediction error, MC.Lasso1+ is preferred because it provides more interpretable weight estimates, as seen in

Figure 8: Weight plots for batch-to-batch comparison when ensemble size is 10 in multiple abrupt changes.

Figures 2 and 6.

Figure 9: Comparison of MC.Lasso1+ and MC.Ridge1+ for (a) single and (b) two change points.

4.2. Real Data Analysis

In this subsection, we apply four methods to the following two real data sets.

Combined Cycle Power Plant (CCPP) Data

This data set from the UCI machine learning repository (<http://archive.ics.uci.edu/ml>) contains 9,568 observations and 5 attributes. The observations were collected in a hourly basis from year 2006 to 2011 when the power plant worked under full load. The objective of the learning is to predict the net hourly electrical energy output. The selected predictors are hourly

average ambient variables *Temperature*, *relative humidity* and *exhaust vacuum*. We divide the data set into 40 data batches of equal sample size with ensemble size 15.

Airplane Data

This data set, originally from Bureau of Transportation Statistics, was published in ASA statistics computing and graphics data expo in the year of 2009. Ikononovska et al. (2009) tested the performance of regression tree based strategy drift to detect and adapt to airplane delay. We demonstrate the performance using 295,998 observations of Northwest Airlines delay information from February to October, 2008 and investigate the concept drift phenomenon in airplane delay. The predictors are *scheduled departure time*, *actual elapsed time*, and *distance between two airports*. For each month we split the data into 4 data batches of equal sample size, giving 36 data batches in total. The ensemble size is chosen to be 25.

In Figure 10, it can be seen that MC.Lasso1+ and the MC.Ridge1+ perform relatively stable and yield slightly smaller RMSE values than the other two methods on most data batches. In Figure 10(a), the RMSE of all four methods show similar prediction accuracy for for CCPP data. In Figure 10(b), the RMSE values differ after batch 26 for Airplane data. In Figure 10(c), we take a closer look at the performance after batch 18, and find that MC.Lasso1+ and MC.Ridge1+ perform better than the other two and adapt quickly in the circumstance of possible concept drift.

The weight plots in Figure 11 illustrate the non-existence of concept drift in the CCPP data. For demonstration purpose, we create the weight plots from the last five data batches (B_{36} - B_{40}) in Figure 11. Observing the flat

Figure 10: RMSE plots for (a) Combined Cycle Power Plant data, (b) for Airplane data, and (c) for Airplane data from batch 18 to 36.

weights in MC.Lasso1+ and MC.WAvg in Figure 11, we do not discover any changes in the instance of power generation process. Although MC.Ridge1+ and MC.Lse1+ yield similar RMSE values in Figure 10 their weight values fluctuate across batches and they are difficult to interpret. In order to justify the estimated weights, we choose two time points (B_{36} and B_{37}), and randomly divide the 15 batches (ensemble size for this example) into two groups at each time point, and draw the density estimate of $\hat{f}(\mathbf{x}_i)$'s in each group in Figure 12. Here, $\mathbf{x}_i$'s belong to the most recent data batch. Both plots show the almost identical densities between two random groups, which supports the equal weights. We try other time points and obtain the same result.

The weight plots of B_{32} - B_{36} in Figure 13 show possible presence of concept drift in the Airplane data. We find considerable changes in the later part of the Airplane data based on the estimated weights of MC.Lasso1+ in Figure

Figure 11: Weight plots for CCPP data.

Figure 12: Density of $\hat{f}$ for CCPP data.

13. We assert that the data batches are not relevant to current base model before batch 28 since the $\hat{w}_1 = \dots = \hat{w}_{17} = 0$ at B_{36} . This suggests that the concept drift happened near the end of July, 2008 (after batch 27) for Northwest Airlines. To support our claim, we choose four time points (B_{32} - B_{35}), divide the 25 batches (ensemble size for this example) into the zero and nonzero weight group using MC.Lasso1+ in Figure 13, and draw the density estimate of $\hat{f}(\mathbf{x}_i)$ in each group in Figure 14. The four density plots clearly show the different distributions of $\hat{f}$ between the zero and nonzero weight group, which justifies the nonzero weights. This example shows that the proposed MC.Lasso1+ effectively tracks concept drift and possesses a grouping effect, as the base models contain clusters of batches with similar weights.

5. Conclusion

The fundamental problem we are interested in is: given a sequential data stream, how should an aggregation-based predictive model be constructed in

Figure 13: Weight plots for Airplane data.

Figure 14: Density of $\hat{f}$ for Airplane data.

order to track time-evolving concepts, make accurate predictions, and produce interpretable aggregation weights? In order to accomplish this goal, we develop penalized regression methods with constraints. Our theoretical investigation of the proposed methods shows that they can adaptively track concept drift through the tuning parameter. Our numerical study shows that the penalized regression approach is preferred to regression or weighted average approaches when concept change occurs. In particular, the penalized regression method with the l_1 penalty quickly adapts to a sudden change, demonstrates a grouping property, and predicts future instances well. However, we also identify some limitations of the proposed approach. For example, our theoretical and empirical studies mostly rely on several assumptions, such as independence, linearity, and equal variance, to simplify the problem. We admit that this might not be realistic in practice, and plan to develop an improved approach that can be robust to deviation from these assumptions. Another limitation concerns the fact that the proposed method provides prediction on a single future instance rather than on a future batch. These limitations should be addressed in a future study.

The proposed methods can be applied to a broad set of applications, such as financial time series, health care, spam classification, and web traffic data. For example, in finance, a study of the relationship between the general mood and individual economic behavior is possible because population sentiment is broadcast every second via social network systems such as Twitter and Facebook. Finding this relationship would have a big impact because public mood analysis from social media offers fast and cost-free measurements on a large scale. Another example is spam classification. Spam is constantly

changing as new kinds emerge over time because spammers change their behaviors to avoid detection, and a learning process should be able to handle this challenge. The proposed methods can be modified to apply to classification problems, and we intend to investigate this in future work.

Appendix A. Appendix

Proof of Theorem 1

Proof. We prove Theorem 1 under the general setting where $P(\mathbf{w}) = \mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w}$ with $\mathbf{D} \mathbf{1}_M = \mathbf{0}$. Given a $L \times M$ $\mathbf{D}$ matrix, let $\mathbf{D} = [\mathbf{D}_{-M}; \mathbf{d}_M]$ where $\mathbf{D}_{-M} = [\mathbf{d}_1, \dots, \mathbf{d}_{M-1}]$ and $\mathbf{w} = (\mathbf{w}_{-M}^T, w_M)^T$ where $\mathbf{w}_{-M}^T = (w_1, \dots, w_{M-1})$ and $w_M = 1 - \sum_{m=1}^{M-1} w_m$. From $\mathbf{D} \mathbf{1}_M = \mathbf{0}_L$, we obtain $\mathbf{d}_M = -\mathbf{D}_{-M} \mathbf{1}_{M-1}$. Then,

$$\begin{aligned} \mathbf{D} \mathbf{w} &= (\mathbf{D}_{-M} - \mathbf{d}_M \mathbf{1}_{M-1}^T) \mathbf{w}_{-M} + \mathbf{d}_M \\ &= \mathbf{D}_{-M} (\mathbf{I}_{M-1} + \mathbf{1}_{M-1} \mathbf{1}_{M-1}^T) \mathbf{w}_{-M} - \mathbf{D}_{-M} \mathbf{1}_{M-1} \\ &= \mathbf{D}_{-M} (\tilde{\mathbf{I}}_{M-1} \mathbf{w}_{-M} - \mathbf{1}_{M-1}) \end{aligned}$$

where $\tilde{\mathbf{I}}_{M-1} = \mathbf{I}_{M-1} + \mathbf{1}_{M-1} \mathbf{1}_{M-1}^T$. For a sufficiently large λ , the first term in (3.1) is negligible. Therefore, we have

$$\mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w} = (\tilde{\mathbf{I}}_{M-1} \mathbf{w}_{-M} - \mathbf{1}_{M-1})^T \mathbf{D}_{-M}^T \mathbf{D}_{-M} (\tilde{\mathbf{I}}_{M-1} \mathbf{w}_{-M} - \mathbf{1}_{M-1})$$

and equate its partial derivative with respect to $w_m, m = 1, \dots, M-1$ to zero, we obtain the equation $\mathbf{w}_{-M} = \tilde{\mathbf{I}}_{M-1}^{-1} \mathbf{1}_{M-1}$. Substituting $\tilde{\mathbf{I}}_{M-1}^{-1} = \mathbf{I}_{M-1} - \mathbf{1}_{M-1} \mathbf{1}_{M-1}^T / M$, we have $\mathbf{w}_{-M} = M^{-1} \mathbf{1}_{M-1}$ and $w_M = 1/M$. $\square$

Proof of Theorem 2

Proof. It suffices to show that $\hat{\mathbf{w}}^T \hat{\mathbf{w}}$ is a decreasing function of λ . The original setting (3.1) can be re-written in matrix notation

$$R_2^{\mathbf{D}}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w}$$

subject to $\mathbf{w} \geq 0$ and $\mathbf{w}^T \mathbf{1}_M = 1$ where $\mathbf{X}$ and $\mathbf{y}$ are defined in (2.1). Here, $\mathbf{w} \geq 0$ indicates that each element of $\mathbf{w}$ is non-negative.

Using the Karush-Kuhn-Tucker (KKT) conditions, for $\boldsymbol{\xi} = (\xi_1, \dots, \xi_M)^T$ and $\beta \geq 0$, we minimize

$$L(\mathbf{w}, \beta, \boldsymbol{\xi}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w} + \beta(\mathbf{w}^T \mathbf{1}_M - 1) - \boldsymbol{\xi}^T \mathbf{w},$$

i.e.

$$\frac{\partial L}{\partial w_k} = 0, \quad \sum_{m=1}^M w_m = 1, \quad \xi_k w_k = 0, \quad \xi_k \geq 0,$$

for $k = 1, \dots, M$. Then,

$$\frac{\partial L}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{D}^T \mathbf{D} \mathbf{w} + \beta \mathbf{1}_M - \boldsymbol{\xi} = 0. \quad (\text{A.1})$$

Multiplying $\mathbf{w}^T$ to the equation (A.1), we have

$$-2\mathbf{w}^T \mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w} + \beta \mathbf{w}^T \mathbf{1}_M - \mathbf{w}^T \boldsymbol{\xi} = 0.$$

Since $\mathbf{w}^T \boldsymbol{\xi} = \sum_{m=1}^M \xi_m w_m = 0$, $\mathbf{w}^T (\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) - 2^{-1} \lambda \mathbf{D}^T \mathbf{D} \mathbf{w} - 2^{-1} \beta \mathbf{1}_M) = 0$, which is equivalent to

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) - \frac{1}{2} \lambda \mathbf{D}^T \mathbf{D} \mathbf{w} - \frac{1}{2} \beta \mathbf{1}_M = 0. \quad (\text{A.2})$$

The solution to equation (A.2) is given by

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \frac{1}{2} \lambda \mathbf{D}^T \mathbf{D})^{-1} (\mathbf{X}^T \mathbf{y} - \frac{1}{2} \beta \mathbf{1}_M),$$

and thus

$$\hat{\mathbf{w}}^T \hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{y} - \frac{1}{2} \beta \mathbf{1}_M)^T (\mathbf{X}^T \mathbf{X} + \frac{1}{2} \lambda \mathbf{D}^T \mathbf{D})^{-2} (\mathbf{X}^T \mathbf{y} - \frac{1}{2} \beta \mathbf{1}_M).$$

□

The proof of Theorem 3

Proof. Note that a point $\mathbf{w}$ achieves a global minimum of the problem (3.2) if and only if all forward directional derivatives at $\mathbf{w}$ are non-negative (see the Chi and Lange (2014)). Thus, at the average probability vector $\mathbf{w}^* = \mathbf{1}_M/M$, the directional vector is

$$\begin{aligned} dL(\mathbf{w}^*) &= \left\langle \sum_{i=1}^n \mathbf{f}_i(\mathbf{f}_i^T \mathbf{w}^* - y_i), \mathbf{v} \right\rangle + \lambda \sum_{l_1 < l_2} |v_{l_1} - v_{l_2}| \\ &= \sum_{m=1}^M (a_m - \bar{a}) v_m + \lambda \sum_{l_1 < l_2} |v_{l_1} - v_{l_2}|, \end{aligned}$$

where $\mathbf{f}_i = \hat{\mathbf{f}}(\mathbf{x}_i)$, $a_m = \sum_{i=1}^n f_{im}(\mathbf{f}_i^T \mathbf{w}^* - y_i)$ for $m = 1, \dots, M$, and $\bar{a} = 1/M \sum_{m=1}^M a_m$. We note that $\sum_m v_m = 0$, and $\sum_{l_1=1}^M (a_{l_1} - \bar{a}) v_{l_2} = 0$ for all $l_2 = 1, \dots, M$. In order to find the lower bound, the Cauchy-Schwarz inequality implies that

$$\begin{aligned} \sum_{l_1=1}^M (a_{l_1} - \bar{a}) v_{l_1} &= \frac{1}{M} \sum_{l_2=1}^M \sum_{l_1=1}^M (a_{l_1} - \bar{a}) v_{l_1} \\ &= \frac{1}{M} \sum_{l_1=1}^M \sum_{l_2=1}^M (a_{l_1} - \bar{a}) (v_{l_1} - v_{l_2}) \\ &= \frac{1}{M} \sum_{l_1=1}^M \sum_{l_2 \neq l_1}^M (a_{l_1} - \bar{a}) (v_{l_1} - v_{l_2}) \\ &= \frac{2}{M} \sum_{l_1 < l_2} (a_{l_1} - \bar{a}) (v_{l_1} - v_{l_2}) \geq -\frac{2}{M} \sum_{l_1 < l_2} |a_{l_1} - \bar{a}| \cdot |v_{l_1} - v_{l_2}|. \end{aligned}$$

Hence, we have

$$dL(w) \geq -\frac{2}{M} \sum_{l_1 < l_2} |a_{l_1} - \bar{a}| \cdot |v_{l_1} - v_{l_2}| + \lambda \sum_{l_1 < l_2} |v_{l_1} - v_{l_2}| \geq 0.$$

Now, it suffices to prove that λ can be taken large so that

$$\lambda \sum_{l_1 < l_2} |v_{l_1} - v_{l_2}| \geq \frac{2}{M} \sum_{l_1 < l_2} |a_{l_1} - \bar{a}| |v_{l_1} - v_{l_2}|.$$

Therefore, any λ which exceeds $\lambda \geq 2/M \max_l |a_l - \bar{a}|$ satisfies the forward directional derivative test. $\square$

Acknowledgement

The second author's work was supported by National Science Foundation (NSF-1021RR193164). The fourth author's work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2007611).

Appendix B. Reference

- Arnold, T. and Tibshirani, R. (2014). Efficient implementations of the generalized lasso dual path algorithm. *Journal of Computational and Graphical Statistics*, doi: 10.1080/10618600.2015.1043010.
- Bartlett, P. L. (1992). Learning with a slowly changing distribution. *Proceeding COLT '92 Proceedings of the fifth annual workshop on Computational learning theory*, pp. 243–252.

- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, **3**, pp. 1-122.
- Chi, E. C. and Lange, K. (2014). Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, doi:10.1080/10618600.2014.948181.
- Chu, F., Wang, Y., and Zaniolo, C. (2004). Mining noisy data streams via a discriminative model. In *Discovery Science*, pp. 47–59.
- Helmbold, D. P. and Long, P. M. (1994). Tracking drifting concepts by minimizing disagreements. *Machine Learning*, **14**, pp. 27–45.
- Ikonomovska, E., Gama, J., Sebastiao, R., and Gjorgjevik, D. (2009). Regression trees from data streams with drift detection. *Lecture Notes in Computer Science*, LNAI 5808, pp. 121–135.
- Kelly, M., Hand, D., and Adams, N. (1999). The impact of changing populations on classifier performance. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM SIGKDD international conference on knowledge discovery and data mining; KDD, pp. 367–371.
- Kim, Seung-JeanKoh, K. S. D. (2009). Trend filtering. *SIAM Review*, **51**, pp. 339–360.
- Klinkenberg, R. (2004). Learning drifting concepts: example selection vs. example weighting. *Intelligent Data Analysis*, **8**, pp. 281–300.

- Klinkenberg, R. and Joachims, T. (2000). Detecting concept drift with support vector machines. In *Machine Learning -International Workshop Then Conference-*, International conference on machine learning, pp. 487–494.
- Kolter, J. Z. and Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, **8**, pp. 2755–2790.
- Kubat, M. and Widmer, G. (1995). Adapting to drift in continuous domains. *Lecture Notes in Computer Science*, **912**, pp. 307–310.
- Kuh, A., Petsche, T., and Rivest, R. L. (1990). Learning time-varying concepts. In *NIPS*, pp. 183–189. Morgan Kaufmann.
- Kuncheva, L. I. (2004). Classifier ensembles for changing environments. In *Lecture Notes in Computer Science*, volume 3077 of *Multiple Classifier Systems; Multiple classifier systems: 5th international workshop, MCS 2004, Cagliari, Italy, June 9-11, 2004 : proceedings*, pp. 1–15.
- Lazarescu, M. M., Venkatesh, S., and Hung H., B. (2004). Using multiple windows to track concept drift. *Intelligent Data Analysis*, **8**, pp. 29–59.
- Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, **3**, pp. 123–131.
- Park, M. Y., Hastie, T., and Tibshirani, R. (2007). Averaged gene expressions for regression. *Biostatistics*, **8**, pp. 212–227.
- Shalev-Shwartz, Shai Singer, Y. K. P.-H. E. (2006). Efficient learning of label

- ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, **7**, pp. 1567–1599.
- Shen, X. and Huang, H.-C. (2010). Grouping pursuit through a regularization solution surface. *Journal of the American Statistical Association*, **105**, pp. 727–739.
- Street, W. N. and Kim, Y. (2001). A streaming ensemble algorithm (sea) for large-scale classification. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 377–382.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, **67**, pp. 91–108.
- Tibshirani, R. and Taylor, J. (2011). The solution path of the generalized lasso. *Annals of Statistics*, 39(3):1335 – 1371.
- Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Technical report.
- Valiant, L. and Waltz, D. (1984). A theory of the learnable. *Communications of the ACM*, **27**, pp. 1134–1142.
- Wang, H., Fan, W., Yu, P., and Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Knowledge discovery and data mining; KDD-2003, pp. 226–235.

- Widmer, G. and Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, **23**, pp. 69–101.
- Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, **5**, pp. 241–259.
- Yang, Y., Wu, X., and Zhu, X. (2005). Combining proactive-reactive prediction for data streams. KDD05, pp. 710–715.
- Yeon, K., Song, M. S., Kim, Y., Choi, H., and Park, C. (2010). Model averaging via penalized regression for tracking concept drift. *Journal of Computational and Graphical Statistics*, **19**, pp. 457–473.