Quilt: Robust Data Segment Selection against Concept Drifts

Minsu Kim, Seong-Hyeon Hwang, Steven Euijong Whang

KAIST

{ms716, sh.hwang, swhang}@kaist.ac.kr

Abstract

Continuous machine learning pipelines are common in industrial settings where models are periodically trained on data streams. Unfortunately, *concept drifts* may occur in data streams where the joint distribution of the data X and label y, $P(X, y)$, changes over time and possibly degrade model accuracy. Existing concept drift adaptation approaches mostly focus on updating the model to the new data possibly using ensemble techniques of previous models and tend to discard the drifted historical data. However, we contend that explicitly utilizing the drifted data together leads to much better model accuracy and propose Quilt, a *data-centric* framework for identifying and selecting data segments that maximize model accuracy. To address the potential downside of efficiency, Quilt extends existing data subset selection techniques, which can be used to reduce the training data without compromising model accuracy. These techniques cannot be used as is because they only assume virtual drifts where the posterior probabilities $P(y|X)$ are assumed not to change. In contrast, a key challenge in our setup is to also discard undesirable data segments with concept drifts. Quilt thus discards drifted data segments and selects data segment subsets holistically for accurate and efficient model training. The two operations use gradient-based scores, which have little computation overhead. In our experiments, we show that Quilt outperforms state-of-the-art drift adaptation and data selection baselines on synthetic and real datasets.

Introduction

Robust AI is becoming important especially in continual learning, which is common in industrial settings where models need to be periodically trained on data streams. The applications include manufacturing, meteorology, finance, and more. Here we assume that *concept drifts* can occur where the joint distribution of the data X and label y, $P(X, y)$ may change over time, leading to decision boundary shifts and model accuracy degradation.

A naive approach of concept drift adaptation is to discard all the historical data, but we would like to retain previous data that is still useful for training. There are many drift adaptation techniques for concept drifts, but most of them take a model-centric approach where they assume that any drifted data is either discarded or replaced with trained models and focus on updating the model or taking an ensemble of previous models to be accurate on the new data. However, throwing away old data is often unacceptable due to heavy investments in data labeling. Even if the knowledge of the data is preserved in the form of models, there are limitations in how much knowledge they retain for making accurate predictions on the new data.

Instead, we contend that taking a *data-centric* approach of selecting data segments to train a model can be a more fundamental solution. We assume that any drift detection technique can be used to identify drift points in the data and thus divide the data into *data segments* using these points. We then formulate the problem of efficiently selecting which data segments result in the best model accuracy when combined with the current (i.e., newest) data segment.

A key challenge for data-centric approaches is efficiency, and we utilize a recent line of data subset selection techniques (Killamsetty et al. 2021b,a) where the goal is to select a minimal subset of the training data using a validation set for training efficiency while obtaining a similar model accuracy as when training on the entire data. However, these techniques assume virtual drifts where the posterior distribution $P(y|X)$ (i.e., the decision boundary) does not change. The assumption makes sense in this problem because any data can be selected to possibly improve model accuracy, and it is a matter of which data is more useful. In comparison, a concept drift setup assumes that the decision boundary may change, which means that some data may negatively affect model accuracy. As a result, we need to solve the more general problem of performing data segment subset selection while discarding data segments with concept drifts.

We then propose a robust data segment selection framework against concept drifts called Quilt for the purpose of improving model accuracy on recent data. Quilt iteratively performs concept drift detection using conventional detection methods and also selects data segments for training the model if a drift occurs. When selecting data segments, Quilt discards data segments with concept drifts using a *disparity* score and also selects a minimal subset of segments without concept drifts such that the model performance is not sacrificed using a *gain* score. Both scores can be computed using gradient values on the training and validation sets with little overhead in computation.

In our experiments, Quilt performs better overall than state-of-the-art drift adaptation baselines on synthetic and real datasets. The benefits are mainly from effectively utilizing previous data segments. In addition, Quilt outperforms existing data-centric concept drift adaptation techniques (Ramírez-Gallego et al. 2017; Dong et al. 2022; Fan 2004) because they do not explicitly evaluate models on selected data segments as Quilt does.

Summary of Contributions: (1) We propose Quilt, a robust data segment selection framework against concept drifts. (2) We design an efficient data subset selection algorithm that holistically selects core data segments while discarding those with concept drifts. (3) We perform extensive experiments on synthetic and real benchmarks and show that Quilt achieves state-of-the-art accuracy and is efficient.

Preliminaries

Concept Drift A concept drift occurs when the statistical properties of a target domain changes arbitrarily (Lu et al. 2019). Suppose we have the time period [0, t] and a sequence of samples $S_{0,t} = \{e_0, \ldots, e_t\}$ where each sample e_i consists of features X_i and a label y_i. If the distribution of $S_{0,t}$ is represented as $P_{0,t}(X, y)$, a concept drift at timestamp $t + 1$ is formally defined as $P_{0,t}(X, y) \neq P_{t+1,\infty}(X, y)$. Here, we consider $S_{0,t}$ as a previous segment and arriving samples from time $t + 1$ as a current segment until the next drift occurs. Note that we may need to look at samples beyond $t + 1$ to actually detect the drift.

The joint distribution can be decomposed into a prior distribution and posterior distribution as follows: $P_t(X, y) = P_t(X) \cdot P_t(y|X)$. While many works relevant to distribution drifts assume that the posterior distribution stays the same (referred to as a virtual drift), we assume more realistic drifts where the posterior distribution may change (referred to as an actual or concept drift).

Data Segments Quilt assumes that the input data stream consists of *data segments* $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ where each segment represents a concept. To identify a data segment, any concept drift detection technique (Lu et al. 2019; Webb et al. 2016; Krawczyk et al. 2017; Gama et al. 2014) can be used. If two concept drifts are detected at t_1 and t_2, we assume the data within the time interval [t_1, t_2] has the same concept and forms a data segment. While some of the previous data segments may benefit the model accuracy on the newest concept, others may even have a negative impact.

Data Subset Selection The goal of data subset selection is efficient learning by taking a minimal subset of the training data while still obtaining similar model accuracy. A common approach is to perform coreset selection, which selects the weighted subsets of data that estimate certain properties of the full data, such as the loss or gradient. More recently, data subset selection frameworks like GLISTER (Killamsetty et al. 2021b) and GRAD-MATCH (Killamsetty et al. 2021a) focus on both efficiency and robustness using a clean held-out validation set. However, most of these works assume that the posterior distribution $P(y|X)$ (i.e., the decision boundary) does not change, which is not true for concept drifts. While Quilt extends these techniques, it solves the more general problem of handling concept drifts.

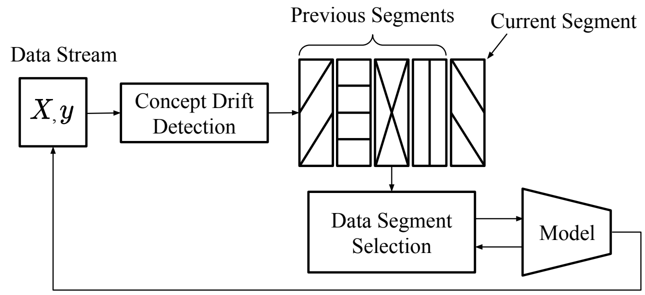


Figure 1: The workflow of Quilt.

Problem Definition

Given an input stream of data segments $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, our goal is to make accurate predictions on the current (i.e., newest) data segment d_N. As with other existing concept drift works, we assume a multi classification setup. We assume the test set d_T is a subset of d_N. For the training and validation sets, a simple setup is to use the previous drifted data only. However, if the previous drifted data is significantly different than d_N, then Quilt will not be effective in selecting data segments that are most suitable for the current data segment. Hence, we take two portions of d_N excluding the test set d_T and add one ($d_N^T \subseteq d_N - d_T$) to the training set, while using the other ($d_N^V = d_N - d_T - d_N^T$) as the validation set. To ensure that the current data segment is large enough for such constructions, we train a model only after at least a certain number of samples (say 100) have been collected since the last concept drift. We then select a subset of the previous data segments $\{d_1, d_2, \ldots, d_{N-1}\}$ that minimizes a trained model's loss on d_N^V when added to the original training set d_N^T. Denoting $L(\theta, S)$ as the loss on a set S using model parameters θ, our problem can be defined as follows:

$$\underset{S \subseteq \{d_1, d_2, \ldots, d_{N-1}\}}{\arg\min} \; L(\underset{\theta}{\arg\min}\, L(\theta, d_N^T \cup S), d_N^V)$$

Quilt Overview

We describe the overall process of Quilt in Figure 1 (the full algorithm is in our technical report (2023)). For each sample from the input data stream, the concept drift detection component checks if there is a concept drift. There are many existing drift detection methods that use the change of data distribution or model performance, and one can plugin any one of them. If there is a drift, a new data segment is created from the drift point and becomes the current data segment. The data segment selection component then selects segments used to update the model (explained in the next section). If there is no drift, the sample is added to the current data segment.

Data Segment Selection

When selecting data segments in the presence of concept drifts, Quilt performs two key operations in a holistic framework: (1) discard data segments that represent drifted concepts compared to the current data segment and (2) select a core subset of data segments that have not drifted for efficient model training without compromising accuracy. The two operations utilize *disparity* and *gain* scores, which are based on gradient values on the training and validation sets and thus have little computation overhead. An additional benefit of using such gradient-based scores is that they are agnostic to the data characteristics. In comparison, statistical distance functions like total variation and Kullback-Leibler divergence are known to have limitations on multivariate data with numeric features and do not scale well to large datasets with high dimensions (Goldenberg and Webb 2019). In the next sections, we explain the gradient computation and elaborate on each score.

Gradient Computation

When computing gradients, we assume neural networks that consist of front layers, which transform the input data to significant embedding features, and a last layer that makes the logit outputs of each class. Let $X'_i \in \mathbb{R}^{d'}$ be the embedding feature of the ith input data X_i with a hidden layer dimension of d', and $z_i \in \mathbb{R}^c$ be the logit outputs computed by $z_i = w \cdot X'_i + b$ using the last layer weights $w \in \mathbb{R}^{d' \times c}$ and bias $b \in \mathbb{R}^c$. To convert a logit z_i into a probability vector $\hat{y}_i$, we use a softmax function: $\hat{y}_i = e^{z_i} / \sum_{j=1}^{c} e^{z_{ij}}$. We can also rewrite the model output $\hat{y}_i$ as a function of the model parameters θ and input data X_i as $\hat{y}_i = f_\theta(X_i)$. With the model output $\hat{y}_i$ and truth label y_i, we compute cross-entropy loss between them as $L_i = L(y_i, \hat{y}_i) = -\sum_{j=1}^{c} y_{ij} \cdot \log(\hat{y}_{ij})$. We use last layer gradient approximation $g = (\nabla_b L, \nabla_w L)$ where gradients of the front layers are not used (Katharopoulos and Fleuret 2018; Ash et al. 2020; Mirzasoleiman, Bilmes, and Leskovec 2020). Using the chain rule, we can compute the gradient of the ith sample as follows: $g_i = (\nabla_b L_i, \nabla_w L_i) = (\hat{y}_i - y_i, (\hat{y}_i - y_i) \cdot X'_i)$.

Disparity Score

We propose a disparity score (abbreviated as $\mathcal{D}$) that measures the dissimilarity between two data distributions and can be used to discard data segments that have concept drifts. Assuming that the data distribution $P(X)$ does not change, a concept drift causes the posterior distribution $P(y|X)$ to change. Hence, a concept drift is similar to how much y changes for the same data. We can capture this notion of disparity in the measure $\mathbb{E}[\|y_t - y_v\|]$, which is the expected amount of label change in a sample where y_t and y_v are truth labels from a training subset and a validation set, respectively. This notion is similar to concept drift severity (Minku, White, and Yao 2010). Directly computing this measure can be expensive where we need to find similar samples in the training and validation sets and measure their label differences. Instead, we define a gradient-based score that is a proxy of this measure and can be computed very efficiently.

Definition 1. The disparity score of a training subset T w.r.t a validation set V is defined as $\mathcal{D}(T, V) = \|\frac{1}{|T|}\sum_{t=1}^{|T|} g_t - \frac{1}{|V|}\sum_{v=1}^{|V|} g_v\| = \|\mathbb{E}[g_t] - \mathbb{E}[g_v]\|$.

The $\mathcal{D}$ score thus measures the L_2-norm distance between two gradient vectors. Intuitively, if a model is fixed, two data segments with similar data distributions should have similar gradients (i.e., low disparity) and vice versa.

We provide a theoretical justification on why the $\mathcal{D}$ score captures concept drift. For analysis purposes, we make the simplifying assumption that the prior distributions of the training and validation sets are the same, although their labels are different. The proof is in our technical report (2023).

Theorem 1. If training subset T and validation set V have the same prior distribution $P_T(X) = P_V(X)$, but different posterior distributions $P_T(y|X) \neq P_V(y|X)$, then $\mathcal{D}(T, V) \leq \mathbb{E}[\|y_t - y_v\|]\sqrt{1 + \sigma^2}$ where $\sigma = \max(\|\mathbb{E}[X']\|)$.

In practice, the prior distribution may change, but we show in our experiments that the $\mathcal{D}$ score is still effective for measuring drifts.

Gain Score

Our data subset selection is based on theoretical foundations of the recent data subset selection literature (Killamsetty et al. 2021b,a). Suppose there exists historical data for training and a validation set. It is known that selecting a data subset whose inner product of the average gradients on the subset and the validation set (called the *gain*) is positive results in a reduction of the model's validation loss at each epoch (Killamsetty et al. 2021b). Intuitively, a gradient vector represents the magnitude and direction of the model parameter updates when performing gradient descent, and it is desirable for the gradients of the training and validation sets to align. Computing the gradient values can be extended to data segments, and we define the *gain* score for data segments (abbreviated as $\mathcal{G}$) as follows:

Definition 2. The gain score of a training subset T w.r.t. a validation set V is $\mathcal{G}(T, V) = \frac{1}{|T|}\sum_{t=1}^{|T|} g_t \cdot \frac{1}{|V|}\sum_{v=1}^{|V|} g_v = \mathbb{E}[g_t] \cdot \mathbb{E}[g_v]$.

Compared to the disparity score, the gain score is less sensitive to concept drifts as it is only affected by the magnitude of the two gradients and angle between them, and not the label differences. Unlike existing data subset selection works where the subset size is set in advance (e.g., top-10% samples), we opt to select all the data segments that have a positive gain score. The reason is that the data segments may contain some levels of concept drift even after discarding the ones with obvious drifts using disparity scores, so we utilize the sign of the gain score to select the useful data segments that can reduce model loss. This issue does not occur if there are no concept drifts, as no data subset is assumed to decrease model performance.

Algorithm

Algorithm 1 shows the data segment selection algorithm of Quilt. We first initialize the model parameters (Step 1). For

Algorithm 1: Data Segment Selection algorithm

Input: Previous data segments $D_{prev} = \{d_1, \ldots, d_{N-1}\}$, training set d_N^T, validation set d_N^V, loss function L, learning rate η, maximum epochs T, disparity threshold T_d

1: Initialize model parameters θ_0
2: **for** epoch t in $[1, \ldots, T]$ **do**
3:     Initialize training subset $S = \emptyset$
4:     $g_V = \frac{1}{|d_N^V|} \sum_{j=1}^{|d_N^V|} g_j$
5:     **for** segment d in D_{prev} **do**
6:         $g_d = \frac{1}{|d|} \sum_{k=1}^{|d|} g_k$
7:         $\mathcal{G}_d = g_d \cdot g_V$
8:         $\mathcal{D}_d = \|g_d - g_V\|$
9:         **if** $\mathcal{G}_d > 0$ and $\mathcal{D}_d < T_d$ **then**
10:             $S = S \cup d$
11:     $S = S \cup d_N^T$
12:     Update $\theta_t = \theta_{t-1} - \eta \frac{1}{|S|} \sum_{e \in S} \nabla_\theta L_e$
13: **return** final model parameters θ_T

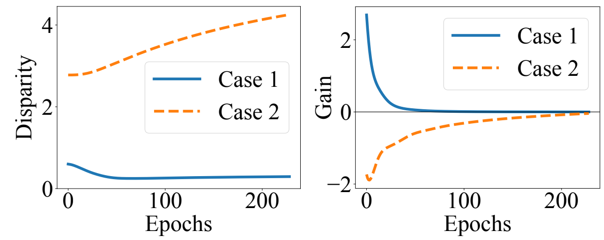
every epoch, we initialize the training subset to an empty set (Step 3). Next, we compute the average gradient on the validation set (Step 4). We then select all previous segments where the gain is positive, and the disparity is sufficiently small (Steps 5–10). Lastly, we add the current training set and finalize the subset for model training (Step 11). We then update the parameters based on the losses and derived gradients of the selected segments (Step 12). After T epochs, we return the final model parameters (Step 13).

Next, we analyze Algorithm 1's complexity. We denote N as the number of data segments and $|S|$ as the average number of selected data segments during T epochs. Let F be the forward pass complexity of the last layer and B the backward pass complexity of all layers. Given one validation set, the complexity is $O((N+1)FT + |S|BT)$, where the first term is for computing the gradients of the data segments and the validation set, and the second term is for updating the model parameters with the selected segments.

Case Study: Two Concepts

In this section, we analyze the behavior of disparity and gain scores with a simple setup where there are only two concepts 0 and 1, and the samples with these concepts have labels 0 and 1, respectively. Let us say the current data segment V has the concept 1, and there is one previous data segment T with a concept of either 0 or 1. We assume that $P_T(X) = P_V(X)$ for simplicity. Since we use the same model state to compute the gradients of T and V, the same prior distribution also leads to the same expected model output $\mathbb{E}[\hat{y}_t] = \mathbb{E}[\hat{y}_v] = (s_1, s_2, \ldots, s_c)$.

Case 1 If T's concept is 1 (i.e., there is no concept drift), we can show that $\mathcal{D}(T, V) \leq 0$ and $\mathcal{G}(T, V) \geq s_1^2 + (s_2 - 1)^2 + \cdots + s_c^2$ (see technical report (2023) for the derivation). We also generate synthetic data for this scenario, shown in our technical report (2023). We monitor the $\mathcal{D}$ and $\mathcal{G}$ scores while running Quilt and observe trends that are consistent



(a) Disparity   (b) Gain

Figure 2: Disparity ($\mathcal{D}$) and gain ($\mathcal{G}$) scores during model training where Case 1 has no concept drift, and Case 2 does. While $\mathcal{G}$ converges to zero as a converged model has nothing to further gain regardless of T's concept, $\mathcal{D}$ does not as it indicates drift severity.

with the derivations above: Figure 2a shows a simulation result where the $\mathcal{D}$ score is empirically close to zero (not exactly zero due to randomness in data sampling), indicating no concept drift; and Figure 2b shows a $\mathcal{G}$ score that is positive and keeps on decreasing as the model fits the concept.

Case 2 If T's concept is 0 (i.e., there is a concept drift), we can show that $\mathcal{D}(T, V) \leq \sqrt{2(1+\sigma^2)}$ and $\mathcal{G}(T, V) \geq s_1(s_1 - 1) + s_2(s_2 - 1) + \cdots + s_c^2$. The derivation is in our technical report (2023). The simulation results in Figure 2a show that the $\mathcal{D}$ score is high compared to non-drift case, indicating a concept drift. In Figure 2b, the $\mathcal{G}$ score is negative and gradually increasing towards zero.

Experiments

We implement Quilt using Python and PyTorch. We evaluate models on separate test sets and repeat all experiments with five different random seeds and write average performances with standard deviations. We use accuracy for evaluation, but also have F_1 score results in our technical report (2023). All experiments are run on NVidia Titan RTX GPUs.

Datasets We evaluate Quilt on four synthetic and five real datasets. Table 1 summarizes the datasets with other experimental settings (see more details in our technical report (2023)). There are four types of concept drifts depending on how the concept changes: sudden (S), gradual (G), incremental (I), and reoccurring (R). That is, a new concept can suddenly, gradually, or incrementally appear, or an old concept can reoccur. The synthetic datasets are designed to have all the different types of concept drifts (Lu et al. 2019). For some real datasets, there is a mixture of types, which we refer to as "Complex."

Model Training We train a simple neural network classifier with cross-entropy loss and an Adam optimizer for all the experiments. We use a periodic holdout evaluation method where, if a concept drift occurs, we first wait for a certain number of samples to arrive to train the model and then perform evaluation. The holdout number depends on the dataset, and we set it to be 10–20% of the size of the average segment size of that dataset. For all the datasets, the

Type	Dataset	Size	#Sgmts	#Ftrs	#Cls	Drift Type
Synthetic	SEA (Street and Kim 2001)	16K	8	3	2	S, R
	Hyperplane (Hulten, Spencer, and Domingos 2001)	16K	8	10	2	G, I
	Random RBF (Bifet et al. 2010)	16K	8	10	5	S, G, I
	Sine (Gama et al. 2004)	16K	8	4	2	S, R
Real	Electricity (Harries 1999)	43.2K	10	6	2	Complex
	Weather (Elwell and Polikar 2011)	18K	10	8	2	Complex
	Spam (Katakis, Tsoumakas, and Vlahavas 2010)	9.3K	9	499	2	G
	Usenet1 & 2 (Katakis, Tsoumakas, and Vlahavas 2008)	1.5K	5	99	2	S, R

Table 1: For each of the nine datasets (four synthetic and five real datasets), we show the total dataset size, the number of data segments, the number of features, the number of classes, and the drift type.

holdout number ranges from 60 to 430. We then use half of this data and historical data for the training set, another half for the validation set, and the rest of the segment for the test set. For a fair comparison with other baselines that do not use validation sets, we make them use both our training and validation sets for their training sets.

Baselines We compare Quilt with four types of baselines:

- **Naïve methods**: *Full Data* uses all the data segments without any selection and *Current Segment* only uses the current segment for training.
- **Model-centric methods**: *HAT* (Bifet and Gavaldà 2009) trains a Hoeffding Adaptive Tree classifier on each sample in an online fashion using the entire data; *ARF* (Gomes et al. 2017) trains an Adaptive Random Forest classifier on each sample using the entire data; *Learn++.NSE* (Elwell and Polikar 2011) takes an ensemble of models trained on previous data segments and adjusts their weights depending on their losses on the current data segment; and *SEGA* (Song et al. 2022) ensembles models trained from equal-length segments of historical data that have the minimum kNN based distributional discrepancy with the current data. We also note that *HAT* and *ARF* have their own mechanisms for detecting concept drifts.
- **Data-centric method**: *CVDTE* (Fan 2004) trains a Cross-Validation Decision Tree Ensemble classifier using individual samples that do not have conflicting predictions between shifted decision boundaries due to concept drifts.
- **Data Subset Selection methods**: *GLISTER* (Killamsetty et al. 2021b) and *GRAD-MATCH* (Killamsetty et al. 2021a) both train a neural network classifier based on data subset selection methods, but with different criteria. *GLISTER* ranks data subsets with gains and selects the top-k subsets with a pre-defined budget. *GRAD-MATCH* simultaneously selects data subsets and adjusts their weights to minimize the gradient error. These baselines are not designed to handle concept drifts.

Parameters For the neural network classifier, we always use one hidden layer with 256 nodes. We set the learning rate to 1e-3 using cross-validation and the number of maximum epochs to 2,000 with early stopping. For each new data segment, we set the disparity threshold using Bayesian optimization with a search interval between (0, 2).

Accuracy and Runtime Results

For each dataset, we evaluate Quilt's accuracy and runtime results for each incoming data segment by setting it as the current segment and then take an average of performances on all the segments. We compare Quilt with the other baselines on six of the datasets as shown in Table 2. The results for the other three datasets are similar and shown in our technical report (2023). Overall, Quilt outperforms all the baselines in terms of accuracy because it effectively utilizes the drifted data. In comparison, *Full Data* is forced to use all the drifted data, while *Current Segment* cannot utilize any historical data. *HAT* performs worse than Quilt because it adaptively learns the recent data without using previous models or data. The three ensemble methods *ARF*, *Learn++.NSE*, and *SEGA* also perform worse. *ARF* can lose useful previous knowledge while replacing an obsolete tree for drift adaptation. Although *Learn++.NSE* and *SEGA* save all or a buffer's worth of past models and uses the current data segment to ensemble them, the models trained from previous data segments have limitations in fitting to the current data segment with simple ensemble techniques. This result shows the advantage of taking a data-centric approach that adaptively selects suitable data segments with explicit model evaluations. *CVDTE* performs worse than Quilt because it simply collects samples that do not have conflicting predictions, regardless of whether they actually benefit model accuracy. For the data subset selection baselines, *GLISTER*'s sample-based selection shows more robust results than *GRAD-MATCH*'s selection of randomly divided batches, but the computation time is significantly greater.

Data Segment Selection Analysis

We next verify whether our data segment selection algorithm actually finds core data segments and discards drifted data segments. For each incoming data segment, we collect all the data segments S that have been selected during the entire training at least once as they all influence the final trained model. In addition, we construct a gold standard solution G by performing an exhaustive evaluation where we evaluate all possible combinations of data segments and choose the one that results in the highest accuracy on the validation set. We then measure the precision and recall of S against G. Finally, we take the average of the precision and recall values

Methods	SEA		Random RBF		Sine		Electricity		Weather		Spam	
	Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
Full Data	.849±.005	3.36	.821±.007	9.43	.449±.032	2.25	.694±.010	7.42	**.800**±.005	4.33	.970±.003	1.17
Current Seg.	.864±.004	0.20	.679±.010	0.56	.899±.004	0.94	.709±.009	0.52	.756±.007	0.26	.955±.003	0.16
HAT	.825±.008	1.38	.514±.010	2.35	.293±.023	1.67	.691±.021	6.43	.729±.011	2.10	.888±.009	25.67
ARF	.825±.008	23.49	.645±.029	44.64	.821±.050	21.40	.713±.011	57.36	.775±.008	30.51	.921±.012	44.83
Learn++.NSE	.804±.005	6.65	.611±.009	5.68	.925±.004	5.73	.698±.004	17.26	.703±.007	7.86	.928±.006	3.81
SEGA	.797±.000	4.37	.825±.000	4.47	.253±.000	4.35	.637±.000	10.26	.777±.000	4.11	.858±.000	6.67
CVDTE	.806±.018	0.02	.614±.015	0.05	.857±.004	0.02	.689±.010	0.04	.731±.009	0.03	.917±.009	0.12
GLISTER	.857±.008	25.89	.794±.014	63.73	.879±.013	14.93	.698±.014	77.46	.793±.010	40.79	.971±.005	14.52
GRAD-MATCH	.853±.009	2.13	.790±.013	6.66	.547±.084	0.80	.686±.015	5.97	.795±.008	3.51	.968±.004	1.13
Quilt	**.888**±.004	2.20	**.833**±.008	3.22	**.936**±.005	4.88	**.728**±.007	5.24	.796±.004	2.00	**.974**±.003	2.59

Table 2: Accuracy and runtime (sec) results on six datasets. We compare Quilt with all the four types of baselines.

Metrics	SEA	Hyperplane	Random RBF	Sine
Precision	0.80	0.75	0.78	0.98
Recall	0.98	1.00	0.96	1.00

Table 3: Comparison of Quilt's selected data segments against the Best Segments results on the synthetic datasets.

for all the segments. We refer to the exhaustive searching method as *Best Segments*.

Table 3 shows the average precision and recall results on the four synthetic datasets, each of which has 8 data segments. An average recall value of 0.96–1.00 suggests that Quilt selects almost all of the useful data segments. The average precision is between 0.75–0.98 because Quilt selects some more similar data segments as well. Interestingly, we observe that these extra segments sometimes improve the model accuracy on the test set as shown in Figure 3, which can happen because the gold standards are selected using the validation set results only. Here we compare Quilt with the *Full Data*, *Current Segment*, and *Best Segments*. As a result, Quilt shows competitive results with *Best Segments* and sometimes outperforms it. We suspect that Quilt's training benefits from the extra data segments during some of its epochs, which improves the model generalization. Hence, Quilt actually achieves most of the room for improvements compared to optimal solutions.

Ablation Study

We perform an ablation study of Quilt to see how the two gradient-based scores contribute to the overall performance. Table 4 evaluates Quilt variants when not using one or both scores for six datasets. The results for the other datasets are similar and shown in our technical report (2023). We compare the accuracy, runtime, the speedup compared to when not using the scores, and the portion of data segments selected (Usage). As a result, removing the disparity score ($\mathcal{D}$) leads to worse accuracy for datasets with severe drifts as Quilt is not able to discard drifted segments. When removing the gain score ($\mathcal{G}$), the accuracy does not decrease

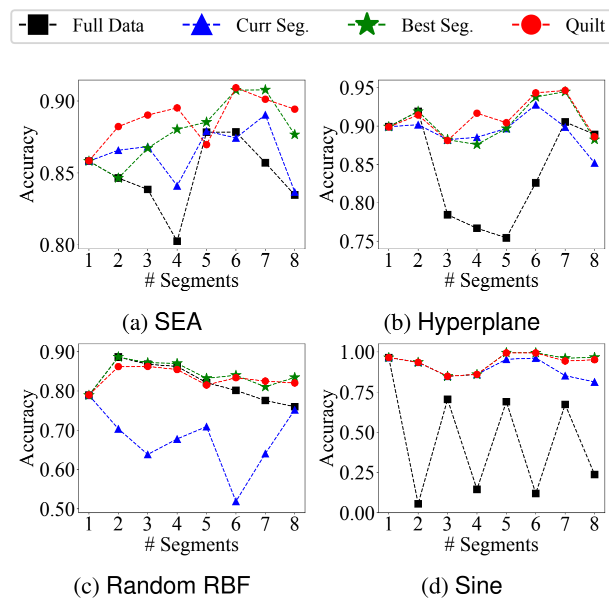


Figure 3: Accumulative model evaluation results against incoming data segments on the four synthetic datasets.

significantly, which is expected, but the runtime worsens because the model training is performed on more data. Removing both scores leads to inaccurate and slow results. For example, on the Random RBF dataset, Quilt is comparable or more accurate than any variant while being 5.1x faster by only using 39.4% of the data segments. While there are some exceptions, Quilt largely provides these benefits for all datasets. Thus, both scores are necessary to obtain both accuracy and efficiency.

Scalability

We show the runtimes of Quilt against the number of data segments on the Random RBF dataset in Figure 4 where we expand the dataset to have 20 segments from 8. As in the accumulative evaluation setting, we assume incoming data segments and measure the runtime for each data seg-

Datasets	Methods	Acc.	Time (Speedup)	Usage
SEA	W/o both	.850±.005	4.49 (1.0×)	100.0%
	W/o $\mathcal{D}$	.890±.004	2.57 (1.7×)	42.2%
	W/o $\mathcal{G}$	.881±.007	3.06 (1.5×)	49.6%
	Quilt	.888±.004	2.20 (2.0×)	30.8%
Random RBF	W/o both	.822±.007	16.51 (1.0×)	100.0%
	W/o $\mathcal{D}$	.828±.008	4.32 (3.8×)	45.3%
	W/o $\mathcal{G}$	.829±.011	9.69 (1.7×)	79.7%
	Quilt	.833±.008	3.22 (5.1×)	39.4%
Sine	W/o both	.444±.032	3.43 (1.0×)	100.0%
	W/o $\mathcal{D}$	.890±.015	2.75 (1.2×)	36.5%
	W/o $\mathcal{G}$	.941±.003	7.77 (0.4×)	23.2%
	Quilt	.936±.005	4.88 (0.7×)	21.1%
Electricity	W/o both	.696±.009	10.71 (1.0×)	100.0%
	W/o $\mathcal{D}$	.711±.013	6.57 (1.6×)	52.9%
	W/o $\mathcal{G}$	.723±.009	6.68 (1.6×)	39.6%
	Quilt	.728±.007	5.24 (2.0×)	27.9%
Weather	W/o both	.798±.006	7.67 (1.0×)	100.0%
	W/o $\mathcal{D}$	.794±.004	2.19 (3.5×)	40.9%
	W/o $\mathcal{G}$	.800±.006	6.85 (1.1×)	75.5%
	Quilt	.796±.004	2.00 (3.8×)	30.7%
Spam	W/o both	.970±.003	4.51 (1.0×)	100.0%
	W/o $\mathcal{D}$	.973±.004	2.31 (2.0×)	52.1%
	W/o $\mathcal{G}$	.972±.002	4.02 (1.1×)	60.5%
	Quilt	.974±.003	2.59 (1.7×)	39.7%

Table 4: Accuracy, runtime (sec), speedup, and data segment usage results of Quilt when not using one or both scores.

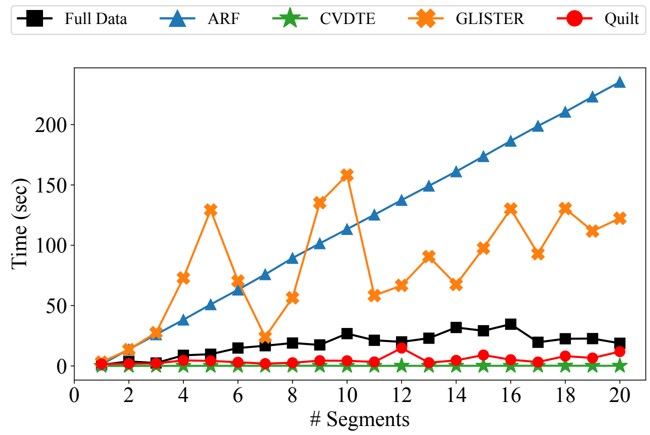


Figure 4: Scalability results of Quilt and baselines on the expanded Random RBF dataset.

ment when setting it to the current data segment. As a result, Quilt scales much better than most baselines because it is able to select small core subsets of segments. Another observation is that adding a data segment may sometimes reduce the overall runtime, which can happen if a new data segment has a smaller core subset than that of the preceding segment.

Related Work

Drift Detection Drift detection (Lu et al. 2019; Webb et al. 2016; Krawczyk et al. 2017; Gama et al. 2014) techniques have been proposed to quantify concept drifts by identifying change points or change time intervals (Basseville and Nikiforov 1993). Drift detection techniques can be largely divided into supervised (e.g., DDM (Gama et al. 2004), EDDM (Baena-Garcıa et al. 2006), and ADWIN (Bifet and Gavaldà 2007)) and unsupervised (e.g., HDDDM (Ditzler and Polikar 2011) and DAWIDD (Hinder et al. 2020)) detection techniques depending on whether a trained model is used. In comparison, Quilt is an orthogonal technique where it takes any input of data segments that can be produced from any of these techniques.

Concept Drift Adaptation Model-centric concept drift adaptation techniques focus on efficiently updating the given model on new data, and previous drifted data is discarded. We categorize these methods as follows: simple re-training of the model; using ensemble models (Gomes et al. 2017; Elwell and Polikar 2011; Song et al. 2022); and adjusting existing models (Bifet and Gavaldà 2009; Domingos and Hulten 2000; Gama, Rocha, and Medas 2003; Hulten, Spencer, and Domingos 2001) depending on the characteristics of the concept drift. In comparison, Quilt does not discard previous data and carefully utilizes it.

Data-centric approaches for concept drift adaptation have been proposed as well. Data reduction techniques (Ramírez-Gallego et al. 2017) clean the data by deleting redundant or noisy samples and features. Drift understanding techniques (Dong et al. 2022) have been proposed where the newest data segment is used as a pattern to filter out obsolete data. The filtering is performed based on comparing the cumulative distribution functions of data, and samples that are filtered out are never re-selected even if they could be beneficial later. The most relevant technique to ours is *CVDTE* (Fan 2004), a sample selection technique where given previous and current models, the goal is to select the samples that do not have conflicting predictions. However, all these techniques have the fundamental limitation that there is no way to see if the data preprocessing results actually improve the model accuracy. In comparison, Quilt takes a more data-driven approach of explicitly evaluating models on selected data segments, while minimizing the computation cost using gradient-based disparity and gain scores.

Conclusion

We proposed Quilt, a data segment selection framework that is robust against concept drifts. Unlike traditional concept drift adaptation frameworks, we take a data-centric approach by directly selecting segments of the training data in a stream that most effectively improves the model accuracy. We design an efficient data subset selection algorithm that selects core data segments while discarding those with concept drifts. The key approach is to use gradient-based disparity and gain scores, which can be used to identify drifts and select useful segments, respectively, and have low computation overheads. Our experiments show how Quilt significantly outperforms both concept drift adaptation and data selection baselines on synthetic and real datasets.

Acknowledgments

This work was supported by SK Hynix AICC (K20.05), by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00157, Robust, Fair, Extensible Data-Centric Continual Learning), and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2022R1A2C2004382). Steven E. Whang is the corresponding author.

References

Ash, J. T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; and Agarwal, A. 2020. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *ICLR*.

Baena-Garcıa, M.; del Campo-Ávila, J.; Fidalgo, R.; Bifet, A.; Gavalda, R.; and Morales-Bueno, R. 2006. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, 77–86. Citeseer.

Basseville, M.; and Nikiforov, I. V. 1993. *Detection of Abrupt Changes: Theory and Application*. ISBN 0131267809.

Bifet, A.; and Gavaldà, R. 2007. Learning from Time-Changing Data with Adaptive Windowing. In *ICDM*, 443–448.

Bifet, A.; and Gavaldà, R. 2009. Adaptive Learning from Evolving Data Streams. In *IDA*, volume 5772, 249–260.

Bifet, A.; Holmes, G.; Kirkby, R.; and Pfahringer, B. 2010. MOA: Massive Online Analysis. *J. Mach. Learn. Res.*, 11: 1601–1604.

Ditzler, G.; and Polikar, R. 2011. Hellinger distance based drift detection for nonstationary environments. In *IEEE CIDUE*, 41–48.

Domingos, P. M.; and Hulten, G. 2000. Mining high-speed data streams. In *KDD*, 71–80.

Dong, F.; Lu, J.; Song, Y.; Liu, F.; and Zhang, G. 2022. A Drift Region-Based Data Sample Filtering Method. *IEEE Trans. Cybern.*, 52(9): 9377–9390.

Elwell, R.; and Polikar, R. 2011. Incremental Learning of Concept Drift in Nonstationary Environments. *IEEE Trans. Neural Networks*, 22(10): 1517–1531.

Fan, W. 2004. Systematic data selection to mine concept-drifting data streams. In *KDD*, 128–137.

Gama, J.; Medas, P.; Castillo, G.; and Rodrigues, P. P. 2004. Learning with Drift Detection. In *SBIA*, volume 3171, 286–295.

Gama, J.; Rocha, R.; and Medas, P. 2003. Accurate decision trees for mining high-speed data streams. In *KDD*, 523–528.

Gama, J.; Zliobaite, I.; Bifet, A.; Pechenizkiy, M.; and Bouchachia, A. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4): 44:1–44:37.

Goldenberg, I.; and Webb, G. I. 2019. Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowl. Inf. Syst.*, 60(2): 591–615.

Gomes, H. M.; Bifet, A.; Read, J.; Barddal, J. P.; Enembreck, F.; Pfahringer, B.; Holmes, G.; and Abdessalem, T. 2017. Adaptive random forests for evolving data stream classification. *Mach. Learn.*, 106(9-10): 1469–1495.

Harries, M. 1999. SPLICE-2 Comparative Evaluation: Electricity Pricing. Technical report.

Hinder, F.; Artelt, A.; Hammer, B.; and Ham, B. 2020. Towards non-parametric drift detection via Dynamic Adapting Window Independence Drift Detection (DAWIDD). In *ICML*, volume 119, 4249–4259.

Hulten, G.; Spencer, L.; and Domingos, P. M. 2001. Mining time-changing data streams. In *KDD*, 97–106.

Katakis, I.; Tsoumakas, G.; and Vlahavas, I. P. 2008. An Ensemble of Classifiers for coping with Recurring Contexts in Data Streams. In *ECAI*, volume 178, 763–764.

Katakis, I.; Tsoumakas, G.; and Vlahavas, I. P. 2010. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowl. Inf. Syst.*, 22(3): 371–391.

Katharopoulos, A.; and Fleuret, F. 2018. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. In *ICML*, volume 80, 2530–2539.

Killamsetty, K.; Sivasubramanian, D.; Ramakrishnan, G.; De, A.; and Iyer, R. K. 2021a. GRAD-MATCH: Gradient Matching based Data Subset Selection for Efficient Deep Model Training. In *ICML*, volume 139, 5464–5474.

Killamsetty, K.; Sivasubramanian, D.; Ramakrishnan, G.; and Iyer, R. K. 2021b. GLISTER: Generalization based Data Subset Selection for Efficient and Robust Learning. In *AAAI*, 8110–8118.

Kim, M.; Hwang, S.-H.; and Whang, S. E. 2023. Quilt: Robust Data Segment Selection against Concept Drifts. Technical report. https://arxiv.org/abs/2312.09691.

Krawczyk, B.; Minku, L. L.; Gama, J.; Stefanowski, J.; and Wozniak, M. 2017. Ensemble learning for data stream analysis: A survey. *Inf. Fusion*, 37: 132–156.

Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; and Zhang, G. 2019. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.*, 31(12): 2346–2363.

Minku, L. L.; White, A. P.; and Yao, X. 2010. The Impact of Diversity on Online Ensemble Learning in the Presence of Concept Drift. *IEEE Trans. Knowl. Data Eng.*, 22(5): 730–742.

Mirzasoleiman, B.; Bilmes, J. A.; and Leskovec, J. 2020. Coresets for Data-efficient Training of Machine Learning Models. In *ICML*, volume 119, 6950–6960.

Ramírez-Gallego, S.; Krawczyk, B.; García, S.; Wozniak, M.; and Herrera, F. 2017. A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, 239: 39–57.

Song, Y.; Lu, J.; Liu, A.; Lu, H.; and Zhang, G. 2022. A Segment-Based Drift Adaptation Method for Data Streams. *IEEE Trans. Neural Networks Learn. Syst.*, 33(9): 4876–4889.

Street, W. N.; and Kim, Y. 2001. A streaming ensemble algorithm (SEA) for large-scale classification. In *KDD*, 377–382.

Webb, G. I.; Hyde, R.; Cao, H.; Nguyen, H.; and Petitjean, F. 2016. Characterizing concept drift. *Data Min. Knowl. Discov.*, 30(4): 964–994.