Evaluation of Updating Strategies for Conformal Predictive Systems in the Presence of Extreme Events

Hugo Werner HUGOWER@KTH.SE
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden
Stena Line, Sweden

Lars Carlsson LARS.CARLSSON@STENALINE.COM
Stena Line, Sweden and Centre for Reliable Machine Learning, University of London, UK

Ernst Ahlberg ERNST.AHLBERG@[STENALINE,GMAIL].COM
Stena Line, Sweden and Department of Pharmaceutical Biosciences, Uppsala University, Uppsala, Sweden

Henrik Boström BOSTROMH@KTH.SE
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden

Editor: Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Giovanni Cherubin

Abstract

Six different strategies for updating split conformal predictive systems in an online (streaming) setting are evaluated. The updating strategies vary in the extent and frequency of retraining as well as in how training data is split into proper training and calibration sets. An empirical evaluation is presented, considering passenger booking data from a ferry company, which stretches over a number of years. The passenger volumes have changed drastically during 2020 due to COVID-19 and part of the evaluation is focusing on which updating strategies work best under such circumstances. Some strategies are observed to outperform others with respect to continuous ranked probability score and validity, highlighting the potential value of choosing a proper strategy.

Keywords: Conformal predictive distributions · Split conformal predictive systems · concept drift

1. Introduction

Today, many companies with less quantitatively focused staff tend to do a lot of manual work. With the increased focus on artificial intelligence, companies are however now aiming for automating many manual steps in their respective businesses. For a ferry company, such as Stena Line, it is important to understand future volumes for the different capacities on board the vessels, in both the long and short term. Understanding volumes of basic capacities such as passengers, cars and freight allows for optimizing staffing both on board and in the port, leading to more efficient operations. These volume estimates play a key role when estimating demand for on board entertainment, shops and restaurants.

Forecasting passenger volumes and optimization of ticket pricing have also been extensively investigated in the airline industry (Littlewood, 2005; Weatherford and Bodily, 1992; McGill and van Ryzin, 1999; Belobaba, 2016) where models are updated at specific decision points described in days before departure. Predictions are typically made using instance based calculations but both linear regressors and time series models have been evaluated. Airlines typically optimise the capacity as the number of seats on an aircraft, and switch aircraft if demand is significantly lower or higher. Due to the low degree of standardisation of vessels and ports coupled with the comparatively high costs of moving vessels that is not an option in the ferry transportation business. This means that there can be significant deviations of on board passenger volumes in the high season and in the low season, for example a ferry that can take 2000 passengers in high season could still sail with less than 50 passengers in low season. Day of week and time of day are two other parameters that influence passenger volumes on board, and volumes tend to fluctuate even short term prior to departure. This requires forecasts that not only result in point predictions but can deliver bounds on uncertainties as well. Since the volume estimates are key components for both efficient scheduling of staff and ordering of required on board consumables, models resulting in well-calibrated probability distributions would allow for efficient operational decision making where uncertainties can be taken into account by decision makers.

Conformal predictive systems (Vovk et al., 2020) generate valid predictive distributions under the assumption that the training and test data is exchangeable. Conformal predictive distributions are useful in decision making contexts as they provide more information than point predictions, as produced by standard regressors, and prediction intervals, as produced by conformal regressors. The original approach to forming predictive distributions was formulated as a transductive approach, incorporating new observations one by one. In theory, such a formulation directly fits an online (streaming) scenario. However, due to that the transductive framework requires training and calibration for each new observation, the computational cost often becomes too high in practical applications. To reduce this cost, the more efficient Split Conformal Predictive Systems (SCPS) have been proposed (Vovk et al., 2020), requiring model training and calibration to take place only once.

The Covid-19 pandemic has had a dramatic impact on most businesses; within the transportation sector the effects have been both positive and negative. On the one hand, people have not been allowed to cross borders due to lock down measures in trying to restrict the spread of the SARS-CoV-2 virus. On the other hand, some other activities, such as internet shopping, have increased, which have driven an increase in transportation of goods. Regardless of the type of business, it is extremely important to be able to react on sudden changes. From a predictive modeling perspective, this highlights the importance of being able to detect or adapt to changes in the underlying data distribution, something which is commonly referred to as *concept drift* (Webb et al., 2016).

In a real streaming scenario, where observations are collected over longer periods of time and concept drift may occur, the choice of strategy to update the predictive model may have a large impact on the resulting predictive performance. For standard (point) predictive models, a large variety of approaches for detecting and adapting to concept drift have been proposed and evaluated, see e.g. (Gama et al., 2014) for an overview. However, for conformal prediction, and conformal predictive systems in particular, such investigations are relatively rare, mainly focusing on detecting violations of the exchangeability assumption rather than adapting to such changes, see e.g. (Fedorova et al., 2012; Volkhonskiy et al., 2017; Vovk et al., 2021). Conclusions drawn from adapting point predictors do not necessarily carry over to conformal predictors, since the latter may not only update the underlying model (point predictor) but also the calibration set. This means that separate investigations of the latter are motivated.

In this study, we will evaluate different strategies for updating conformal predictive systems in the context of streaming data, and in particular investigate their effectiveness in the presence of extreme changes in the underlying data distribution.

The remainder of this paper is structured as follows. In Section 2, we restate the definition of Split Conformal Predictive Systems (Vovk et al., 2020), outline the algorithm that will be used for detecting concept drift (Simple Jumper) (Vovk et al., 2021) and describe six strategies for updating the underlying model and calibration set. In Section 3, we describe the experimental setup and present the results. Finally, in Section 4, we summarize the main findings of the investigation and point out directions for future research.

2. Methods

We first recapitulate split conformal predictive systems and the Simple Jumper algorithm for detecting violations of the exchangeability assumption. We then define the six different strategies that will be evaluated. Finally, we present the performance metrics that will be used in the evaluation.

2.1. Conformal predictive systems

Conformal predictive systems for regression output well-calibrated cumulative probability distributions over the possible target values. In this paper, we utilize a computationally efficient variant, called Split Conformal Predictive System (SCPS), which was introduced in (Vovk et al., 2020).

Let $\{z_1, z_2, \ldots, z_n\}$ be a set of observations, where $z_i = (x_i, y_i)$. We assume a scenario where the conformal predictive system is given a set of m observations for proper training, i.e.,: $\{z_1, \ldots, z_m\}$ and $n - m$ corresponding examples for calibration. An outline of SCPS is shown in Algorithm 1. Here, $\hat{y}$ is a prediction of a machine learning algorithm using the proper training set and $\hat{\sigma}$ is an estimate of the quality of $\hat{y}$. In the outline the conformity function used is defined as:

$$A_m(z_1, \ldots, z_m, (x, y)) := \frac{y - \hat{y}}{\hat{\sigma}}, \tag{1}$$

which yields

$$C_i = \hat{y} + \frac{\hat{\sigma}}{\hat{\sigma}_{m+i}}(y_{m+i} - \hat{y}_{m+i}), i = 1, \ldots, n - m \tag{2}$$

which is directly used in Algorithm 1. The predictive distribution is then defined as:

$$Q(z_1, \ldots, z_n, (x, y), \tau) := \begin{cases} \frac{i+\tau}{n-m+1} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{1, 1, \ldots, n-m\}, \\ \\ \frac{i'+1+(i''-i'+2)\tau}{n-m+1} & \text{if } y = C_{(i)} \text{ for } i \in \{1, \ldots, n-m\}, \end{cases} \tag{3}$$

where $m < n$, $i' := \min\{j|C_{(j)} = C_{(i)}\}$, $i'' := \max\{j|C_{(j)} = C_{(i)}\}$ and $\tau \sim \mathcal{U}(0,1)$ is independently sampled for each y_i.

Algorithm 1 Split Conformal Predictive System

```
Input: Proper training set $\{z_1, \dots, z_m\}$, calibration set $\{z_{m+1}, \dots, z_n\}$, test object $x$.
for $i = 1, \dots, n-m$ do
    $C_i = \hat{y} + \frac{\hat{\sigma}}{\hat{\sigma}_{m+i}}(y_{m+i} - \hat{y}_{m+i})$.
end
Sort $C_i$ in increasing order such that $C_{(1)} \leq \ldots \leq C_{(n-m)}$
Set $C_0 = -\infty$ and $C_{n-m+1} = \infty$
Return the predictive distribution $Q$.
```

2.2. Conformal Martingales

Changes in the generating distribution can result in that the predictive ability of the model is reduced over time. In settings where the generating distribution is expected to change, detecting such changes becomes very important. One approach to detect changes is to apply a betting martingale such as the Simple Jumper, described in (Vovk et al., 2021), and outlined in Algorithm 2. When Algorithm 2 returns an S_i greater than a predefined threshold, one may assume that a change has occurred.

Algorithm 2 Simple Jumper

```
Input: p-values $(p_1, p_2, \ldots)$, parameter $J$
$f_\epsilon(p) = 1 + \epsilon(p - 0.5)$
$C_{-1} = C_0 = C_1 = 1/3$
$C = 1$
for $i = 1, 2, \ldots$ do
    for $\epsilon \in \{-1, 0, 1\}$ do
        $C_\epsilon = (1 - J)C_\epsilon + (J/3)C$
    end
    for $\epsilon \in \{-1, 0, 1\}$ do
        $C_\epsilon = C_\epsilon f_\epsilon(p_i)$
    end
    $S_i = C = C_{-1} + C_0 + C_1$
end
Return $(S_1, S_2, \ldots)$
```

2.3. Updating strategies

We assume a scenario where the conformal predictive system is given an initial sequence of n training examples (objects with labels), i.e., $(z_1 = (x_1, y_1), \ldots, z_n = (x_n, y_n))$, and then evaluated on k test examples, where for each test example $z_{n+i} \in \{z_{n+1}, \ldots, z_{n+k}\}$. The conformal predictive system is assumed to have access to the object x_{n+i} but not the label y_{n+i}.

All strategies considered here, Strategies 1-6, for forming conformal predictive systems split the initial sequence into a proper training and calibration set, which are given as input to Algorithm 1. The six strategies that are considered in this study vary in how each subsequent observation is handled, e.g., how the respective proper training and calibration sets are updated.

Strategy 1 (S1) The initial split conformal predictive system (SCPS) is kept unchanged and thus new data is discarded.

```
Input: training sequence $Z_t = (z_1, \dots, z_n)$, test sequence $(z_{n+1}, \dots, z_{n+k})$
randomly split $Z_t$ into $Z_p$ and $Z_c$
for $i = 1, \dots, k$ do
    $Q_i = \text{SCPS}(Z_p, Z_c, x_{n+i})$
    evaluate $Q_i$ w.r.t. $y_{n+i}$
end
```

Strategy 2 (S2) The underlying model of the SCPS is kept unchanged, while the calibration set is replaced every fth test observation with the l latest observations. In essence, the calibration set is updated using a sliding window of size l, moving f observations at each update.

```
Input: training sequence $(z_1, \dots, z_n)$, test sequence $(z_{n+1}, \dots, z_{n+k})$, parameters $f$ and $l$
$Z_p = (z_1, \dots, z_{n-l})$
$Z_c = (z_{n-l+1}, \dots, z_n)$
for $i = 1, \dots, k$ do
    if $i \mod f = 0$ then
        $Z_c = (z_{n+i-1-l}, \dots, z_{n+i-1})$
    end
    $Q_i = \text{SCPS}(Z_p, Z_c, x_{n+i})$
    evaluate $Q_i$ w.r.t. $y_{n+i}$
end
```

Strategy 3 (S3) A new SCPS is produced every fth observation, using all observed examples, which are split randomly into a proper training and a calibration set.

```
Input: training sequence $Z_t = (z_1, \dots, z_n)$, test sequence $(z_{n+1}, \dots, z_{n+k})$, parameter $f$
randomly split $Z_t$ into $Z_p$ and $Z_c$
for $i = 1, \dots, k$ do
    if $i \mod f = 0$ then
        randomly split $Z_t$ into $Z_p$ and $Z_c$
    end
    $Q_i = \text{SCPS}(Z_p, Z_c, x_{n+i})$
    $Z_t = Z_t + z_{n+i}$
    evaluate $Q_i$ w.r.t. $y_i$
end
```

Strategy 4 (S4) A new SCPS is produced every fth example, where the l most recent examples are split randomly into a proper training and calibration set.

```
Input: training sequence $Z_t = (z_1, \dots, z_n)$, test sequence $(z_{n+1}, \dots, z_{n+k})$,
parameters $f$ and $l$
$Z_t = (z_{n-l+1}, \dots, z_n)$
randomly split $Z_t$ into $Z_p$ and $Z_c$
for $i = 1, \dots, k$ do
    if $i \mod f = 0$ then
        $Z_t = (z_{n+i-l}, \dots, z_{n+i-1})$
        randomly split $Z_t$ into $Z_p$ and $Z_c$
    end
    $Q_i = \text{SCPS}(Z_p, Z_c, x_{n+i})$
    evaluate $Q_i$ w.r.t. $y_i$
end
```

Strategy 5 (S5) Similar to S2, but the calibration set is only updated when a change is detected (using SimpleJumper). The size of the updated set is not predetermined, but contains the largest number of the most recent examples for which SimpleJumper will not detect a change.

```
Input: training sequence $Z_t = (z_1, \dots, z_n)$, test sequence $(z_{n+1}, \dots, z_{n+k})$,
parameters $m$ and $J$
randomly split $Z_t$ into $Z_p$ and $Z_c$
$s = 1$
for $i = 1, \dots, k$ do
    $Q_i = \text{SCPS}(Z_p, Z_c, x_{n+i})$
    evaluate $Q_i$ w.r.t. $y_{n+i}$
    $p_i = Q_i(y_{n+i})$
    $S_{1,i} = \text{SimpleJumper}((p_s, \dots, p_i), J)$
    if $S_{1,i} > 100$ then
        $j = 0$
        $S_{2,0} = \text{SimpleJumper}((p_i), J)$
        while $S_{2,j} < 100$ & $i - j \geq s$ do
            $j = j + 1$
            $S_{2,j} = \text{SimpleJumper}((p_i, \dots, p_{i-j}), J)$
        end
        $Z_c = (z_{n+i-j}, \dots, z_{n+i})$
        $s = i$
    end
end
```

2.4. Evaluation

To evaluate the overall performance of the strategies, we use the same loss function as in (Vovk et al., 2020), namely Continuous Ranked Probability Score (CRPS). CRPS measures the squared error of a cumulative distribution function compared to the oracle, a step

Strategy 6 (S6) Similar to S5, but where both the calibration and the proper training set are updated when a change is detected.

```
Input: training sequence $Z_t = (z_1, \dots, z_n)$, test sequence $(z_{n+1}, \dots, z_{n+k})$,
parameters $m$ and $J$
randomly split $Z_t$ into $Z_p$ and $Z_c$
$s = 1$
for $i = 1, \dots, k$ do
    $Q_i = \mathrm{SCPS}(Z_p, Z_c, x_{n+i})$
    evaluate $Q_i$ w.r.t. $y_{n+i}$
    $p_i = Q_i(y_{n+i})$
    $S_{1,i} = \mathrm{SimpleJumper}((p_s, \dots, p_i), J)$
    if $S_{1,i} > 100$ then
        $j = 0$
        $S_{2,0} = \mathrm{SimpleJumper}((p_i), J)$
        while $S_{2,j} < 100$ & $i - j \geq s$ do
            $j = j + 1$
            $S_{2,j} = \mathrm{SimpleJumper}((p_i, \dots, p_{i-j}), J)$
        end
        $Z_c = (z_{n+i-j}, \dots, z_{n+i})$
        $Z_p = (z_1, \dots, z_{n+i-j-1})$
        $s = i$
    end
    $Q_i = \mathrm{SCPS}(Z_p, Z_c, x_{n+i})$
    evaluate $Q_i$ w.r.t. $y_{n+i}$
end
```

function at the label. Let F be the cumulative distribution and y_i the label for instance i then CRPS is defined as:

$$\mathrm{CRPS}(F, y_i) = \int_{-\infty}^{\infty} (F(y) - \mathbf{1}_{y>y_i})^2 dy. \tag{4}$$

To determine if the methods produce valid probabilistic distributions we look at the distribution of the labels location on our cumulative distribution functions generated for the evaluation dataset. For valid strategies, the cumulative probabilities for the observed target values in the test set should be uniformly distributed between zero and one. This means that for a chosen probability on the distribution function the proportion of labels which in fact are smaller than the corresponding value should be equal to the probability. Let $\alpha_i = F(y_i)$, i.e. α_i is the value of the probability distribution at the point of the label. Then, to generate a calibration plot we calculate the percentage of α that are smaller or equal to the values $\{0, 0.01, 0.02, \dots, g_i, \dots, 1\}$. For valid system where there are N α $\frac{1}{N}|j : \alpha_j \leq g_i| = g_i$, i.e. the proportion of α which are smaller or equal to a value g_i is g_i. To quantify the performance from the calibration plots the L^2 norm of the distance from the theoretical line, is calculated for each strategy, where $L^2(d) = \sqrt{d_1^2 + \dots d_n^2}$ and $d_i = \frac{1}{N}|j : \alpha_j \leq g_i| - g_i$.

3. Experiments

In this section, we first describe the experimental setup and then the results of the experiment.

3.1. Experimental setup

The experiments concern Stena Line passenger booking data for all departures on one route between 2017 and 2020. Due to COVID-19 there was a significant drop in the number of passengers in 2020. To simulate passenger volumes going back to normal in 2021 without generating synthetic data the available data was reordered. The data from 2017 was moved to follow 2020 and relabeled as 2021, see Figure 1. The features of the data set are summarized in Table 1. The target variable, NO_OF_BOARDED_GUESTS, is the number of guests that had boarded at the time of departure.

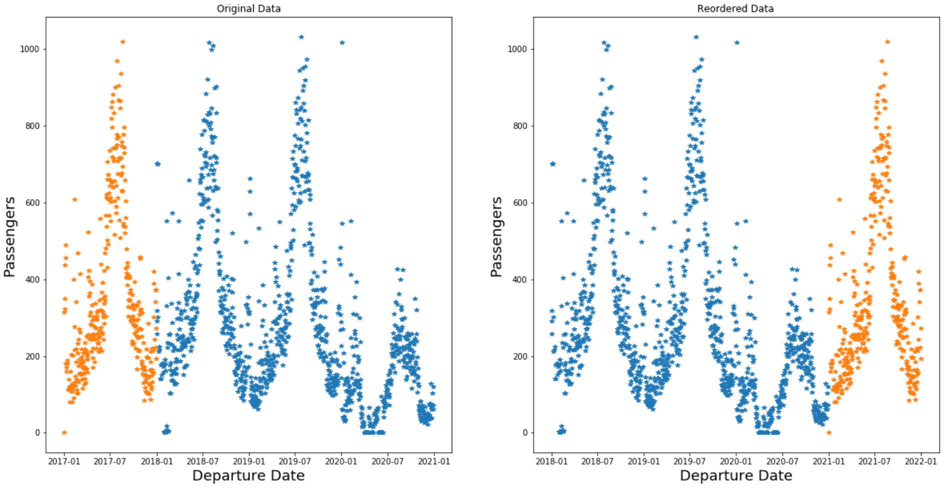


Figure 1: Original data and reordered data.

For each strategy presented in Section 2.3, the experiment was run with and without the booking state, i.e., with and without the features NO_OF_BOOKINGS and NO_OF_GUESTS, which represent the number of bookings and number of guests, respectively, two weeks before the departure. The features used in both cases were WEEKDAY, MONTH, WEEKEND and WEEK and when the booking state was used the features NO_OF_BOOKINGS and NO_OF_GUESTS were also included. Each individual experiment was run 10 times and the average results are reported.

The parameters for the strategies were selected as follows; $n = 712$ such that initial data sequence contained the samples from 2018 and 2019, i.e. $z_i, i \in \{1, 2 \ldots, n = 712\}$. The updating frequency parameter was set to $f = 7$ to mimic a weekly update. In strategy 2, to isolate the effects of only changing the data samples in the calibration set and not the sizes, the size of the sliding window was set to $l = 214$. In strategy 4 the size of the sliding window was set to $l = 712$ in order to show the effect of changing the data samples available for training but avoid the effect of changing the number available samples.The threshold for detecting a change in Algorithm 2 was set to $S_i > 100$ which is in line with (Vovk et al., 2021). In Algorithm 1 the quality, $\hat{\sigma}$ of the predictions, $\hat{y}$ was not estimated, the parameter was set as $\hat{\sigma} = 1$. The parameter J in Algorithm 2 was set to $J = 0.01$. The underlying model used was Random Forest (Pedregosa et al., 2011), which was only retrained when a strategy updated the proper training sequence.

Feature	Description
WEEKDAY	Day of the week encoded as integers, {0, 1, . . . , 6}
MONTH	Month as an integer {1, 2, . . . , 12}
WEEKEND	1 if departure is in on Saturday or Sunday, 0 otherwise {0,1}
WEEK	Week number, {1, 2, . . . , 52}
NO_OF_BOOKINGS	Number of unique bookings for the departure
NO_OF_GUESTS	Number of passengers booked for the departure
NO_OF_BOARDED_GUESTS	Number of passengers boarded at the time of departure

Table 1: Features included in data set.

3.2. Experimental results

For each strategy, CRPS scores were calculated for the entire evaluation data set as well as for the first and second year separately. The averages of the CRPS scores from the ten runs with booking state are presented in Table 2 and without booking state in Table 3. The lowest (best) value in each column is highlighted with bold-face. Table 4 and Table 5 show the L^2 norm of the deviation in the calibration plots for the runs with and without booking state, respectively.

Figure 2 includes a calibration plot for each strategy with and without booking state. Each plot is generated from the outputs of the 10 runs for the given strategy and feature set. Each sub figure shows a calibration plot computed from the output from year 2019 (Y1), 2020 (Y2) and both 2019 and 2020 (Y1+Y2).

Table 2: CRPS with the booking state

Strategy	Y1+Y2	Y1	Y2
$S1$	13.64	13.21	14.08
$S2$	15.32	17.19	**13.45**
$S3$	**12.55**	**11.29**	13.80
$S4$	13.39	12.28	14.49
$S5$	14.15	13.59	14.71
$S6$	13.09	11.92	14.28

Table 3: CRPS without booking state

Strategy	Y1+Y2	Y1	Y2
$S1$	47.38	59.03	35.70
$S2$	48.80	54.65	42.96
$S3$	33.15	30.92	**35.38**
$S4$	33.49	28.59	38.39
$S5$	41.15	44.29	37.95
$S6$	**30.82**	**25.61**	36.15

Table 4: L^2-norm with booking state

Strategy	Y1+Y2	Y1	Y2
$S1$	0.57	0.69	0.47
$S2$	0.18	0.28	**0.17**
$S3$	0.21	0.51	0.33
$S4$	**0.08**	0.32	0.32
$S5$	0.19	**0.15**	0.37
$S6$	0.47	0.42	0.79

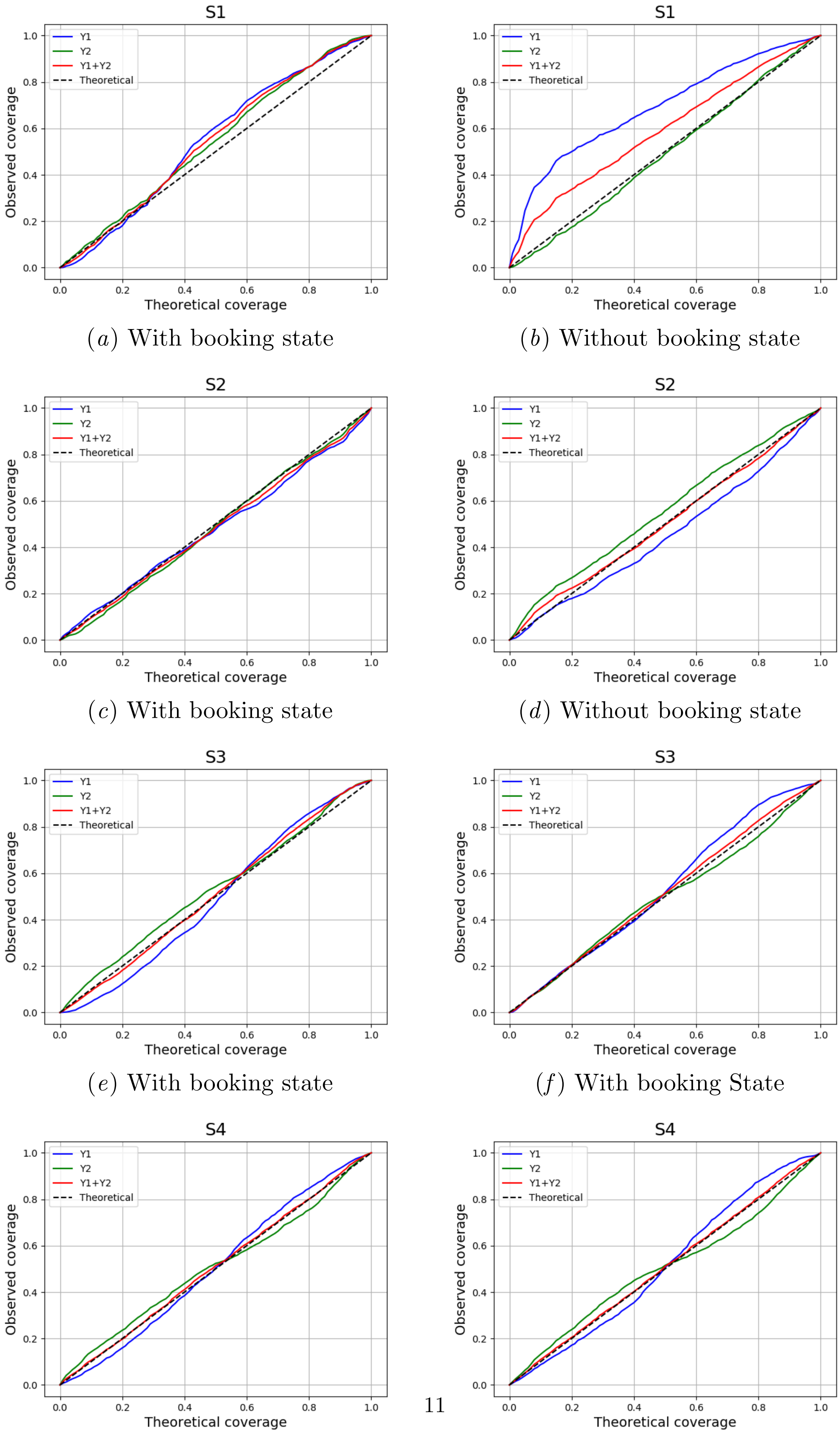
Table 5: L^2-norm without booking state

Strategy	Y1+Y2	Y1	Y2
$S1$	0.98	2.08	**0.17**
$S2$	0.16	0.55	0.55
$S3$	0.15	0.47	0.23
$S4$	**0.07**	0.43	0.40
$S5$	0.63	0.87	0.53
$S6$	0.74	**0.40**	1.13

3.3. Discussion

Which strategy performs the best apparently depends on the performance metric and the dataset. When CRPS is used as a metric, strategy 3 appears favorable performing well for both 2020 and 2021, and feature sets. However, if the L^2-norm is used, strategy 2 may be preferred in the case when the booking state is included but when it is not strategy 4 would perhaps be preferred. In general the differences between the best methods are rather small.

When comparing the results in Table 2 and 3, it is clear that the overall performance, as measured by CRPS, drops substantially when the booking state is not used. This may



(*a*) With booking state

(*b*) Without booking state

(*c*) With booking state

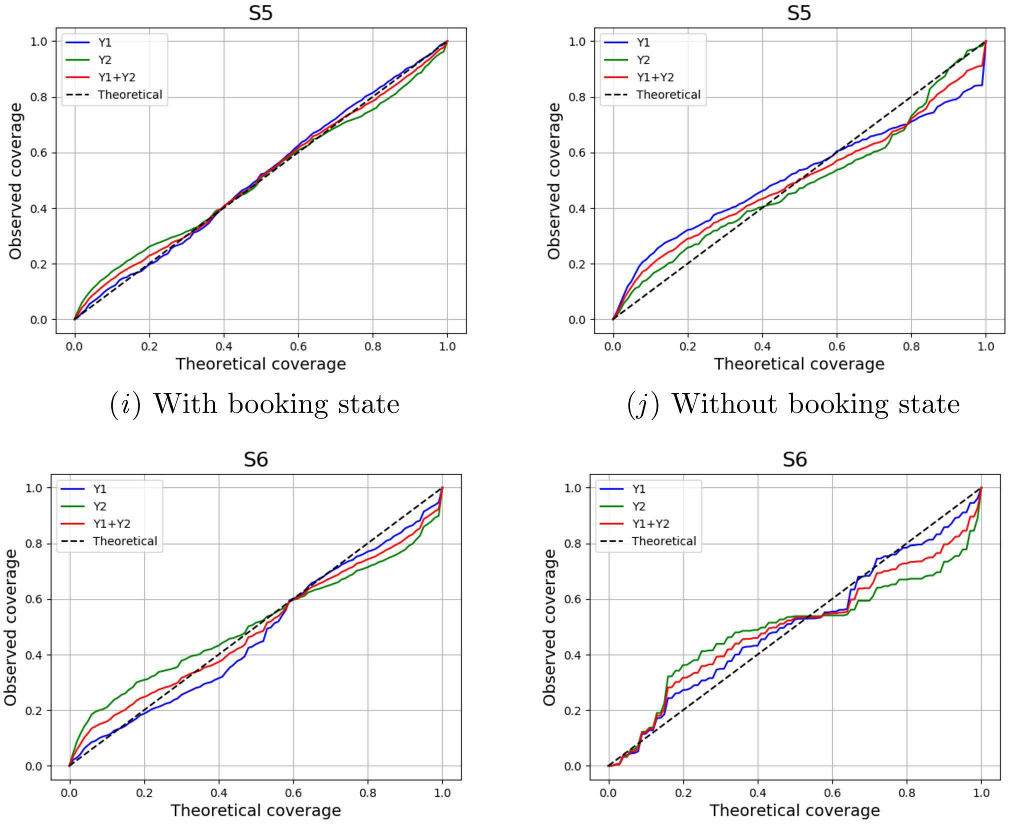
(*d*) Without booking state

(*e*) With booking state

(*f*) With booking State

(*g*) With booking state

(*h*) Without booking state



(i) With booking state

(j) Without booking state

(k) With booking state

(l) Without booking state

Figure 2: Calibration plots for the evaluated strategies with and without booking state.

be explained by the fact that the booking state two weeks before a departure is highly correlated to the number of passengers at the departure, which makes the prediction task easier. When the passenger volumes decrease, the correlation between the booking state and the number of passengers at departure still remains. This may also explain why the L^2 norms are lower for most strategies when the booking state is included as can be seen in Table 4 and 5. When looking the L^2 norms, the combined value for 2020 and 2021 (Y1+Y2), appears to be better than both 2020 (Y1) and 2021 (Y2). This is however due to a skew in the output from 2020 and 2021 which cancel each other out, in 2020 the predictions are too high and in 2021 the predictions are too low. Which leads to a skew in the predictive distributions, clearly exemplified by Figure $2(d)$. In the experiments, the updates for strategies S2, S3 and S4 were done quite often (every seventh observation), but similar results were achieved by using Algorithm 2 to determine when an update was needed, resulting in much fewer updates.

4. Concluding remarks

The presented experimental results show that employing a strategy for updating split conformal predictive systems can significantly improve the performance, as measured by continuous-ranked probability score (CRPS) and L^2-norm of the deviation in the calibration. No clear winner among the strategies was observed in the experiment. However, updating the underlying model and calibration set with the most recent data is better than not updating the model and calibration set, independently of what dataset is considered. In addition, we see that the application of martingales to detect concept drift, as implemented by the Simple Jumper algorithm, can clearly reduce the amount of retraining required, hence reducing the computational cost required for keeping high performing models available in the online setting.

There are several possible directions for further research. One obvious direction involves conducting more large-scale empirical investigations, covering additional datasets and learning algorithms as well as concept drift of different types, e.g., with less abrupt data distribution shifts than what has been considered here. Another direction concerns developing and evaluating more complex updating strategies, e.g., including tuning of the employed hyperparameters, such as the threshold for the Simple Jumper algorithm. An important direction for future research includes evaluating the strategies in real decision making contexts, e.g., by calculating the actual profit from using the different strategies.

Acknowledgment

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. HB was partly funded also by the Swedish Foundation for Strategic Research (CDA, grant no. BD15-0006).

References

P.P. Belobaba. Optimization models in rm systems: Optimality versus revenue gains. *Journal of Revenue and Pricing Management*, 15(3):229–235, Jul 2016.

V. Fedorova, A. Gammerman, I. Nouretdinov, and V. Vovk. Plug-in martingales for testing exchangeability on-line. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL http://icml.cc/2012/papers/808.pdf.

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

K. Littlewood. Special issue papers: Forecasting and control of passenger bookings. *Journal of Revenue and Pricing Management*, 4(2):111–123, 2005.

J. McGill and G. van Ryzin. Revenue management: Research overview and prospects. *Transportation Science*, 33(2):233–256, 1999. doi: 10.1287/trsc.33.2.233. URL https://doi.org/10.1287/trsc.33.2.233.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

D. Volkhonskiy, E. Burnaev, I. Nouretdinov, A. Gammerman, and V. Vovk. Inductive conformal martingales for change-point detection. In *Conformal and Probabilistic Prediction and Applications*, pages 132–153. PMLR, 2017.

V. Vovk, I. Petej, I. Nouretdinov, V. Manokhin, and A. Gammerman. Computationally efficient versions of conformal predictive distributions. *Neurocomputing*, 397:292–308, 2020.

V. Vovk, I. Petej, I. Nouretdinov, E. Ahlberg, L. Carlsson, and A. Gammerman. Retrain or not retrain: Conformal test martingales for change-point detection. *CoRR*, abs/2102.10439, 2021. URL https://arxiv.org/abs/2102.10439.

L. Weatherford and S. Bodily. A taxonomy and research overview of perishable-asset revenue management: Yield management, overbooking, and pricing. *Operations Research*, 40(5): 831–844, 1992. doi: 10.1287/opre.40.5.831. URL https://doi.org/10.1287/opre.40.5.831.

G.I. Webb, R. Hyde, H. Cao, H.L. Nguyen, and F. Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016.