Conformal Stacked Weather Forecasting

Jelmer Neeven J.NEEVEN@STUDENT.MAASTRICHTUNIVERSITY.NL
Evgueni Smirnov SMIRNOV@MAASTRICHTUNIVERSITY.NL
Department of Data Science and Knowledge Engineering, Maastricht University, Maastricht 6200MD, The Netherlands

Abstract

In this paper we propose to apply the stacking method to aggregating multi-output predictions from different weather-forecasting domains (websites). Depending on the aggregating procedure (non-conformal/conformal), the results can be bare multi-output predictions or multi-output prediction regions. The experiments show the applicability of the stacking method on real data related to eight weather-forecasting domains.

Keywords: Conformal Prediction, Multi-Output Prediction, Stacking, Weather Forecasting

1. Introduction

Among the many services offered on the Internet, there are hundreds of websites that provide online predictions in different areas. This includes for example weather forecasting, predicting sport results, stock forecasting etc. While the quality of predictions may vary between different websites, an intriguing question is whether they can be aggregated into more accurate predictions, and if so, whether prediction regions can be built.

In this paper we provide an answer to the question imposed above in the context of weather forecasting. Given a set of weather-forecasting domains (websites), we propose a method that aggregates the multi-output predictions from these domains into a new multi-output prediction. Our method is essentially a stacking method (Wolpert, 1992), hence the name stacked weather forecasting (SWF). The method first collects meta data that consists of the multi-output predictions from the weather-forecasting domains and the true weather measurements for a certain period of time. It then trains a meta multi-output predictor on the meta data, mapping the multi-output predictions from the weather-forecasting domains to new aggregated multi-output predictions. The experiments show that the aggregated predictions are more accurate than any of the individual predictions; i.e. we can successfully combine weather-forecasting domains.

To compute regions over the aggregated multi-output predictions we introduce conformal stacked weather forecasting (CSWF) which is essentially SWF employing conformal multi-output regression models (Shafer and Vovk, 2008). The experiments show that CSWF provides (almost) valid prediction regions for each weather variable, although it is difficult to evaluate their quality. To judge the latter precisely we would need to compare them to the prediction regions of the weather-forecasting websites which are not (publicly) available.

The methods of SWF and CSWF are related to research in weather forecasting modeling (Bauer et al., 2015). To reduce the generalization complexity and the error of the weather-

forecasting models different types of ensemble methods have been proposed (Zhang and Pu, 2010). In essence, these methods are averaging methods, i.e. they average the outputs of the weather forecasting models. To ensure model diversity, the ensemble methods either use different types of weather forecasting models or use randomization injection when weather forecasting models are of the same type (analogously to Bagging (Breiman, 1996)). The weather-forecasting ensembles differ from the methods of SWF and CSWF proposed in this paper in terms of:

- aggregation procedure: the weather-forecasting ensembles employ averaging while SWF and CSWF can employ any type of multi-output regression models (including averaging);
- model access: the weather-forecasting ensembles do have access to the models they aggregate while SWF and CSWF do not. This is an important distinction, as it allows SWF and CSWF to be applied (in theory) to any set of variables for which multiple predictions are available (e.g. through online services).

The rest of the paper is organized as follows. The task of weather forecasting is formalized in Section 2. Section 3 and 4 introduce the SWF method and CSWF method, respectively. The process of data gathering is given in Section 5, and the experiments are described in Section 6. Section 7 concludes the paper.

2. Task of Weather Forecasting

Let X be an input space and $Y \subseteq \mathbb{R}^M$ be an output space defined by M output weather variables $Y_m \subseteq \mathbb{R}, m \in \{1, 2, \ldots, M\}$. A weather forecast domain W is defined as a tuple consisting of a labeled space $X \times Y$ and a probability distribution P over $X \times Y$. Any training instance is a tuple $(x_d, y_{d+l}) \in X \times Y$ where x_d represents measurements on day d and y_{d+l} represents the true values of the weather variables for day $d + l$ for positive integers d and l. The training data set T for lth day predictions is a multi set of instances $(x_d, y_{d+l}) \in X \times Y$ drawn from the distribution P. Given an unlabeled test instance $x_d \in X$, *the weather forecasting task for the domain* W is to find a lth-day prediction $\hat{y}_{d+l} \in Y$ for x_d according to P.

To solve the weather forecasting task, we identify a multi-output regression model h from a hypothesis space H of models h ($h : X \to Y$). We identify h as that model in H that best fits the training data T according the search strategy employed. Given an unlabeled test instance $x_d \in X$, the multi-output regression model h provides a lth-day prediction $\hat{y} \in Y$ for x_d.

3. Stacked Weather Forecasting

We assume the existence of S weather forecast domains W_s for $s \in \{1, 2, \ldots, S\}$. Any two domains W_{s_1} and W_{s_2} ($s_1 \neq s_2$) can have different input spaces X_{s_1} and X_{s_2} but do share the same output space Y. This implies that the multi-output regression models h_s for different weather forecast domains W_s can be different but they do provide predictions in the same output space Y.

We assume that for any weather forecast domain W_s we do not have access to its multi-output regression model h_s. However, we do assume that for any weather forecast domain W_s we have access to the predictions of W_s for all the weather variables. Formally, this means that there is a validation data set $V_s \subseteq (X_s \times Y)$ generated from the probability distribution P_s related to W_s. For all the weather forecast domains W_s the validation data sets V_s are indexed sets with the same index set D of days; i.e. the validation sets have the same size and for each day $d \in D$ there exists instance $(x_{d,s}, y_{d+l,s}) \in V_s$ for all $s \in \{1, 2, \ldots, S\}$.

The weather forecast domains W_s handle the validation data sets V_s. For any domain W_s an instance $(x_{d,s}, y_{d+l,s}) \in V_s$ is tested by the multi-output regression model h_s. This means that the model h_s provides a lth-day prediction $\hat{y}_{d+l,s} \in Y$ for the unlabeled instance $x_{d,s}$. For any day $d \in D$ we create a meta instance $\{\hat{y}_{d+l,s}\}_{s\in\{1,2,\ldots,S\}}$ with true output y_{d+l} for day $d + l$ [1]. The new meta instances connect the estimations $\hat{y}_{d+l,s} \in Y$ and the true outputs y_{d+l} for all the days in D and they form a meta training data set T. We train a meta multi-output regression model f from a hypothesis space F of models f ($f : Y^S \to Y$) using the meta data T. The meta model f provides a prediction $\hat{y}_{d+l} \in Y$ for any unlabeled test instance $\{\hat{y}_{d+l,s}\}_{s\in\{1,2,\ldots,S\}}$ for $d \notin D$. This means that if we know the predictions $\hat{y}_{d+l,s} \in Y$ from all the models h_s using the meta model f we can aggregate them into the lth-day prediction $\hat{y}_{d+l} \in Y$.

The pseudocode for our proposed stacked weather forecasting method is provided in Algorithm 1.

Algorithm 1 SWF: Stacked Weather Forecasting

Input: Index set D of days,
Number l for the days to be predicted,
Number S of weather forecast domains,
Predictions $\hat{y}_{d+l,s}$ for day $d + l$ from any day $d \in D$ and
any domain W_s with $s \in \{1, 2, \ldots, S\}$,
True outputs y_{d+l} for day $d + l$ for any $d \in D$.
Output: Meta multi-output regression model f.

```
Set the meta training data set T equal to ∅;
for all d ∈ D do
    Add meta instance ({ŷ_{d+l,s}}_{s∈{1,2,...,S}}, y_{d+l}) to the meta training data set T;
end for
Train meta multi-output regression model f on the meta training data set T;
return f.
```

The method of stacked weather forecasting is a model-independent method; any type of multi-output regression model can be used for the final meta model. In our experiments we have used a multi-output k-nearest neighbor regression model (MkNNR). Given a test instance, MkNNR first finds the k nearest neighbors of the test instance in the input space. Then, it provides a value for each output variable Y_m ($m \in \{1, \ldots, M\}$) estimated

1. We note that the true output y_{d+l} is the same for all $s \in \{1, \ldots, S\}$; i.e. for all $s \in \{1, \ldots, S\}$ we have $y_{d+l} = y_{d+l,s}$.

as the weighted average of the values for Y_m among the k nearest neighbors. By construction, MkNNR can be viewed as a set of M single-output k-nearest neighbor regressions (SkNNR$_m$) that share the search for the k nearest neighbors. This property is used when we discuss how to implement a conformal extension of MkNNR in the next Subsection.

4. Conformal Stacked Weather Forecasting

Conformal stacked weather forecasting (CSWF) is the SWF method that trains a conformal multi-output regression model f on the meta training data set T. Since conformal multi-output regression models are not available, in this paper we propose a straightforward multi-output extension of the conformal single-output k-nearest neighbor regression (CSkNNR) (Papadopoulos et al., 2011). The extension, which we call conformal multi-output k-nearest neighbor regression (CMkNNR), is a set of M CSkNNR$_m$ models, one for each weather variable Y_m. The CSkNNR$_m$ models share the search for the k nearest neighbors to reduce the computational complexity of CMkNNR. In this way, given a test instance, each CSkNNR provides a predictive region Γ_m for its corresponding weather variable Y_m and the set of all the prediction regions Γ_m forms the final output of the CMkNNR model.

We note that both the MkNNR model and CMkNNR model do not explicitly handle the dependencies that might exist between the output variables which can definitely reduce the generalization performance of these models. However, due to the lack of conformal multi-output regression models, we do not have currently an alternative.

5. Data Gathering and Preprocessing

To test our method of stacked weather forecasting we have selected eight weather forecast websites. These are domains that are listed on Google when searching for "weather forecast" and provide weather predictions in a way that can easily be scraped. The weather forecasting models used by these domains are not public; i.e. the only information we have are the weather predictions provided by these models. The selected domains and the abbreviations used in the sections hereafter are as follows:

- BB: bbc.com
- TD: timeanddate.com
- WO: weatheronline.co.uk
- WT: weather.com
- AW: accuweather.com
- HW: holiday-weather.com
- ZV: zoover.nl
- WF: weather-forecast.com

As these websites provide forecasts for many locations, a manual selection of 24 locations has been made (see Table 1), aimed to be roughly evenly spread[2] across the United States. The chosen locations have been limited to the United States because of the difficulty of accessing ground-truth weather records. The latter have been provided by the National Climatic Data Center (NCDC) and National Oceanic and Atmospheric Administration (NOAA), both American organizations (Menne et al., 2012).

Table 1: Locations for weather forecast

Los Angeles (CA)	Flagstaff (AZ)	Raleigh (NC)
Yuma (AZ)	Las Vegas (NV)	Cleveland (OH)
Tucson (AZ)	Fresno (CA)	Rochester (NY)
San Antonio (TX)	Sacramento (CA)	Bangor (ME)
Pensacola (FL)	Salt Lake City (UT)	Chicago (IL)
Miami (FL)	Kansas City (MO)	Minneapolis (MN)
Atlanta (GA)	Nashville (TN)	Boise (ID)
Albuquerque (NM)	Knoxville (TN)	Portland (OR)

Due to the limited information provided by the NCDC records, only five continuous weather variables have been selected: maximum temperature (°C), minimum temperature (°C), average temperature (°C), wind speed (km/h), and precipitation (mm). Therefore, for any day and any location we have a training instance of which:

- the output consists of the true values for the five variables provided by NCDC, and
- the input consists of the estimated values for the five variables provided by each domain; i.e. 5×8 number of values in total (see line 3, Algorithm 1).

The data has been collected for next-day weather forecasting ($l = 1$) and seven-day weather forecasting ($l = 7$). That is, if an instance output contains true values for a day $d + l$, then its inputs contain predictions provided by the weather forecasting domains around $12:00$ on day d. In total, data has been gathered for 24 locations over 97 consecutive days, resulting in 2240 instances for the next-day forecasts and 2097 instances for seven-day forecasts. Both the scraped predictions and ground truth values (obtained through the NOAA API) have been made publicly available to facilitate use in future research[3]. After collection, this data has been preprocessed. Where necessary, values have been converted to the metric system, and missing values have been processed as follows:

- If the average temperature is missing, it is replaced by the average of the minimum and maximum temperature.

2. Originally, 53 locations had been selected with a distance 150km to 450km in between. However, this selection was then reduced to the 24 listed locations, as only those are covered by all of the selected forecast websites. The distances between locations in this subset has not been evaluated separately, so they may no longer be evenly spread.
3. https://github.com/jneeven/Weather-Forecasting-Data

- As five of the eight websites do not provide any predictions for precipitation, these five features have been removed from the input data (i.e. each input instance contains 35 values).

As the predicted variables have different ranges, the input values have been normalized using min-max normalization (described in (Al Shalabi et al., 2006)) such that they are in range $[0, 1]$. The latter is important for estimating plausible distances on the input space, especially when we use k-nearest neighbor regression models. Since the precipitation data contains many outliers, these values have been replaced by their square root before applying normalization. This has helped decrease the width of the prediction regions of CSWF somewhat, although they are still quite large as can be seen in Section 6.3.2.

6. Experiments

This section presents our experimental set-up (Subsection 6.1), validation procedures (Subsection 6.2), results, and analysis (both in Subsection 6.3).

6.1. Experiment Setup

6.1.1. Setup for Stacked Weather Forecasting

The multi-output k-nearest neighbor regression model for stacked weather forecasting has been implemented in Python using the scikit-learn library. The model uses the Euclidean distance metric and calculates its estimations of the output variables by weighted interpolation of the values associated with the nearest neighbors. The model has been optimized with an internal validation process and values of 8 and 13 have been found optimal for the parameter k in the case of next-day and seven-day forecasting, respectively.

6.1.2. Setup for Conformal Stacked Weather Forecasting

The conformal multi-output k-nearest neighbor regression model for conformal stacked weather forecasting has been coded in Matlab using the conformal single-output k-nearest neighbor regression implementation from (Papadopoulos et al., 2011). The model has been initialized analogously to the multi-output k-nearest neighbor regression model (see previous subsection).

6.2. Validation

6.2.1. Measures for Generalization Performance

The generalization performance of stacked weather forecasting has been estimated using normalized root mean squared error (NRMSE) which is obtained by normalizing predictions and target values of each output weather variable over the total range of that variable in all target values, again using min-max normalization to range $[0, 1]$. As the models predict five output weather variables simultaneously, this results in five error values per test instance. To simplify the comparison of errors, only the mean of the five (normalized) error values is reported.

The generalization performance of conformal stacked weather forecasting has been estimated using two measures for each output weather variable given a significance level ϵ. The first measure is the error e estimated as the proportion of the instances of which the true value is outside of the predictive region for the variable. The second measure is the width w of the predictive region for that variable.

6.2.2. Validation Procedures

Two types of validation procedures have been employed: k-fold cross validation and sliding window. The k-fold cross-validation procedure has been set with k equal to 10. It has been employed under the assumption that the data has been i.i.d. generated. We note that this assumption does not always hold due to the similarity of weather forecast predictions in a series of consecutive days with a relatively stable weather.

The second validation procedure, sliding window, has been employed, since it is more natural for the types of applications considered in the paper. In this procedure, a model is trained on the data from day d_i till day d_j (referred to as the *window* for $d_i < d_j$) and then predicts the output weather variables for day $d_j + l$ with $l > 0$. When predicting for the following day, $d_j + 1 + l$, the model is retrained from scratch using a new window from day $d_i + 1$ till day $d_j + 1$ (i.e. the window "slides" over the available data with one day). Hence, the window always consists of n instances where $n = d_j - d_i$. In our experiments n has been set to 65 days, resulting in 32 testing days. The sliding-window procedure has been employed under the assumption that the data has been under the exchangeability assumption (Shafer and Vovk, 2008). We note that this assumption does not always hold for weather data, for example for days that belong to different seasons.

For both validation procedures, we have naively assumed that the weather forecasting models of the websites presented in Section 5 had also been trained and tested according to those procedures [4]. In this way we can compare the generalization performance of stacked weather forecasting with that of these models.

6.3. Results

This section provides experimental results and conclusions which are first given for stacked weather forecasting and then for conformal stacked weather forecasting.

6.3.1. Results for Stacked Weather Forecasting

Table 2 shows the results for all the weather forecasting domains (websites) and stacked weather forecasting (SWF) based on multi-output k-nearest neighbor regression. The results are given in terms of the averaged normalized root mean squared errors for the five output weather variables: maximum temperature, minimum temperature, average temperature, wind speed, and precipitation. The errors have been estimated using the cross-validation procedure and sliding window procedure described in Subsection 6.2.

The table shows that SWF has the best generalization performance: its averaged NMRSE is significantly lower than those of all eight weather forecasting domains in all

4. For example in case of the 10-fold cross-validation procedure, when we test the first fold of predictions made by a model, we assume that this model has been trained on the data of the next 9 folds.

Table 2: Average and standard deviation of normalized root mean squared errors for the weather forecasting domains and stacked weather forecasting (SWF) based on multi-output k-nearest neighbor regression. Numbers in bold indicate statistically better results than any other domain obtained with paired t-tests on a significance level of 0.05.

	BB	TD	WO	WT	AW	HW	ZV	WF	SWF
Cross-validation (next day)	0.048 ±0.027	0.035 ±0.020	0.063 ±0.050	0.045 ±0.027	0.036 ±0.022	0.055 ±0.035	0.054 ±0.029	0.053 ±0.028	**0.026** ±0.016
Sliding window (next day)	0.049 ±0.028	0.037 ±0.020	0.062 ±0.051	0.045 ±0.026	0.038 ±0.024	0.057 ±0.034	0.052 ±0.028	0.054 ±0.030	**0.031** ±0.022
Cross-validation (seven days)	0.061 ±0.034	0.063 ±0.039	0.085 ±0.054	0.065 ±0.036	0.056 ±0.031	0.075 ±0.040	0.069 ±0.035	0.071 ±0.040	**0.046** ±0.028
Sliding window (seven days)	0.066 ±0.037	0.069 ±0.044	0.090 ±0.059	0.070 ±0.039	0.059 ±0.035	0.081 ±0.042	0.071 ±0.037	0.076 ±0.046	**0.053** ±0.034

experiments. The obtained results allow us to conclude that the SWF method is a good method for combining weather forecasts from different domains. The method is capable of significantly reducing the weather forecasting error.

Analyzing the reasons for the success of the SWF method is a difficult problem (Wolpert, 1992). Here we just illustrate one of the main reasons, namely the diversity of the weather forecasting domains that the method combines. For that purpose we have first calculated the normalized Euclidean distance between the predictions of the domains and SWF, and then applied the multidimensional scaling (MDS) technique (Kruskal, 1964). The result is a two-dimensional map with stress of 0.036 and mean distance of 0.121 presented in Figure 1. The map shows that the weather forecasting domains are rather diverse in terms of the weather forecasts which can be one of the explanations why SWF reduces significantly the forecast error. Analyzing the position of the SWF method, we observe that SWF is a border method on the MDS map; i.e. it is different from others. However, two of its closest neighbors are the best two weather forecasting domains which suggests that we can get similar results by combining only those two domains. A similar situation is shown for the seven-day predictions in Figure 2 which has a stress of 0.048 and a mean distance of 0.137, although the position of SWF has moved slightly towards the center.

6.3.2. Results for Conformal Stacked Weather Forecasting

Figure 3 and Tables 3 and 4 show the results for conformal stacked weather forecasting (CSWF) using the cross-validation and sliding window procedures when making next-day predictions. The results are given in terms of the error e and the width w of the predicted regions for each of the five output weather variables: maximum temperature, minimum temperature, average temperature, wind speed, and precipitation, at significance level $\epsilon \in [0.0, 1.0]$.

These results clearly show that the regions predicted in the case of cross validation are (conservatively) valid. The width of the predicted regions varies; as expected, the widths of the maximum temperature and minimum temperature are bigger than that of the averaged temperature. In the case of sliding window, the predicted regions are almost valid. This

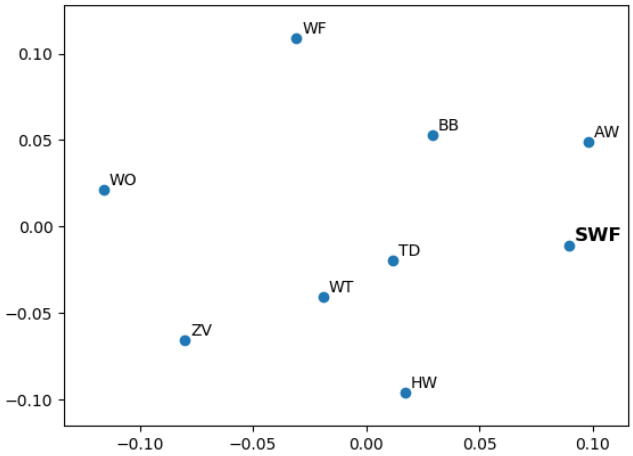

Figure 1: MDS plot of weather forecasting diversity for next-day predictions.

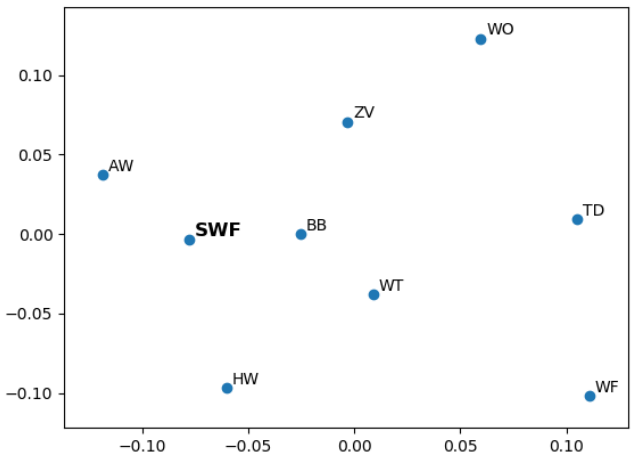

Figure 2: MDS plot of weather forecasting diversity for seven-day predictions.

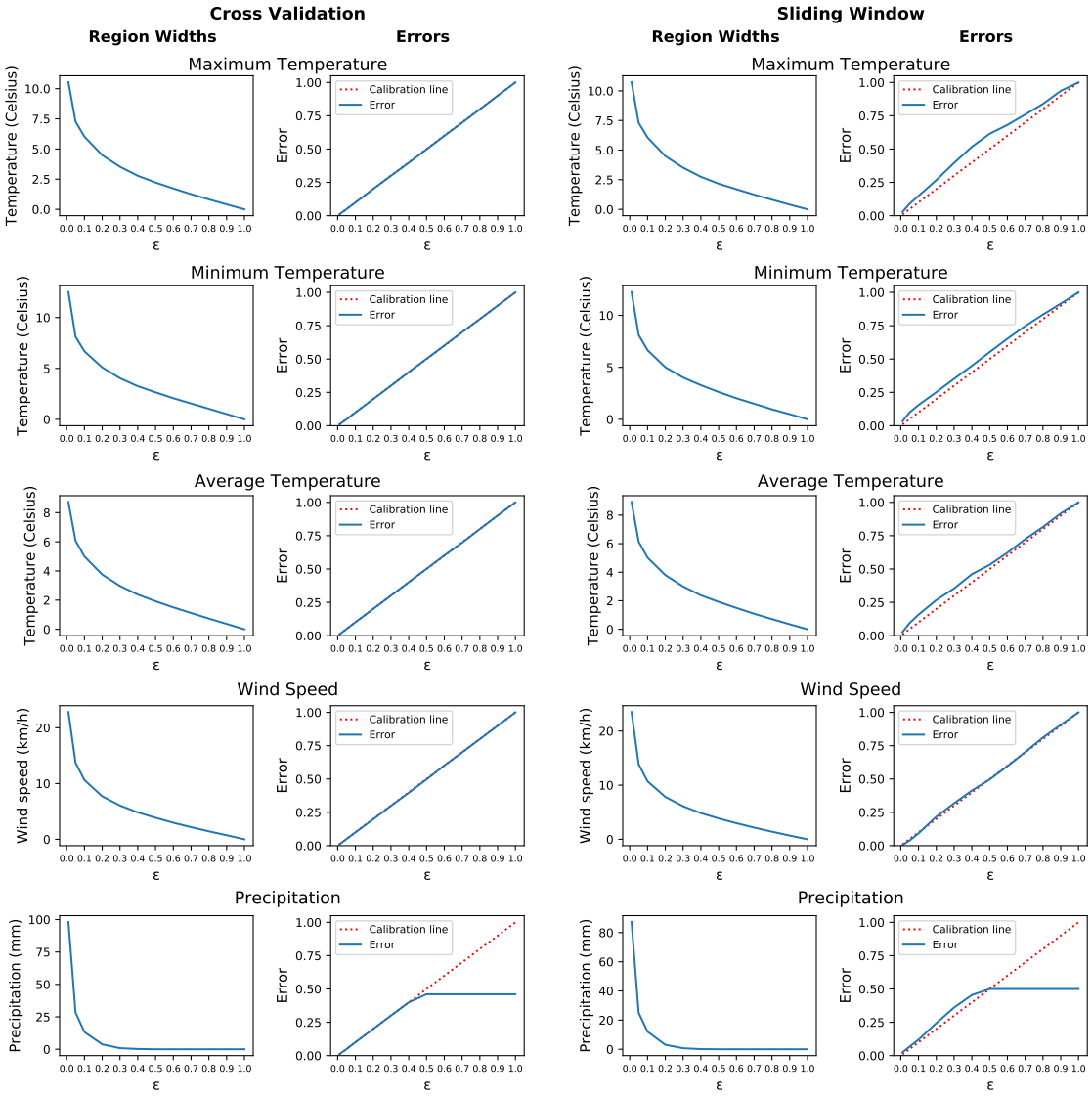

Figure 3: Error and width plots of CSWF predicting the next day based on cross validation (left) and sliding window (right).

is eventually due to the fact that the exchangeability assumption does not always for the data. The width of the predicted regions varies analogously to the case of cross validation.

Figure 4 and Tables 5 and 6 present the results of CSWF for seven-days prediction based on the cross-validation and sliding window procedures. Similarly to the next-day predictions, the regions are valid in the case of cross validation and almost valid in the case of sliding window. For all output variables the predicted regions are larger than those of next-day predictions which correspond to the larger error values observed in Table 2.

Table 3: Errors and widths of the prediction regions of CSWF predicting the next day based on cross validation.

	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$		$\epsilon = 0.2$		$\epsilon = 0.3$		$\epsilon = 0.4$		$\epsilon = 0.5$		$\epsilon = 0.6$		$\epsilon = 0.7$		$\epsilon = 0.8$		$\epsilon = 0.9$		$\epsilon = 1$	
	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w
Max. Temp.	0.0	10.5	0.0	7.3	0.1	6.0	0.2	4.5	0.3	3.5	0.4	2.8	0.5	2.2	0.6	1.7	0.7	1.3	0.8	0.8	0.9	0.4	1.0	0.0
Min. Temp.	0.0	12.5	0.0	8.1	0.1	6.7	0.2	5.1	0.3	4.0	0.4	3.3	0.5	2.7	0.6	2.1	0.7	1.6	0.8	1.0	0.9	0.5	1.0	0.0
Av. Temp.	0.0	8.7	0.1	6.1	0.1	5.0	0.2	3.8	0.3	3.0	0.4	2.4	0.5	1.9	0.6	1.5	0.7	1.1	0.8	0.7	0.9	0.4	1.0	0.0
Wind speed	0.0	22.8	0.0	13.7	0.1	10.6	0.2	7.7	0.3	6.1	0.4	4.8	0.5	3.9	0.6	3.0	0.7	2.2	0.8	1.4	0.9	0.7	1.0	0.0
Precipitation	0.0	98.1	0.0	28.3	0.1	13.3	0.2	3.8	0.3	0.9	0.4	0.2	0.5	0.0	0.5	0.0	0.5	0.0	0.5	0.0	0.5	0.0	1.0	0.0

Table 4: Errors and widths of the prediction regions of CSWF predicting the next day based on sliding window.

	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$		$\epsilon = 0.2$		$\epsilon = 0.3$		$\epsilon = 0.4$		$\epsilon = 0.5$		$\epsilon = 0.6$		$\epsilon = 0.7$		$\epsilon = 0.8$		$\epsilon = 0.9$		$\epsilon = 1$	
	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w
Max. Temp.	0.0	10.7	0.1	7.3	0.1	6.1	0.3	4.5	0.4	3.5	0.5	2.7	0.6	2.2	0.7	1.7	0.8	1.2	0.8	0.8	0.9	0.4	1.0	0.0
Min. Temp.	0.0	12.2	0.1	8.1	0.2	6.7	0.3	5.0	0.4	4.0	0.4	3.3	0.6	2.6	0.7	2.0	0.7	1.5	0.8	1.0	0.9	0.5	1.0	0.0
Av. Temp.	0.0	8.9	0.1	6.1	0.2	5.0	0.3	3.8	0.4	3.0	0.5	2.4	0.5	1.9	0.6	1.5	0.7	1.1	0.8	0.7	0.9	0.4	1.0	0.0
Wind speed	0.0	23.5	0.0	13.9	0.1	10.7	0.2	7.8	0.3	6.1	0.4	4.8	0.5	3.9	0.6	3.0	0.7	2.2	0.8	1.4	0.9	0.7	1.0	0.0
Precipitation	0.0	87.3	0.1	25.1	0.1	12.1	0.2	3.1	0.4	0.7	0.5	0.1	0.5	0.0	0.5	0.0	0.5	0.0	0.5	0.0	0.5	0.0	1.0	0.0

Table 5: Errors and widths of the prediction regions of CSWF predicting seven days ahead based on cross validation.

	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$		$\epsilon = 0.2$		$\epsilon = 0.3$		$\epsilon = 0.4$		$\epsilon = 0.5$		$\epsilon = 0.6$		$\epsilon = 0.7$		$\epsilon = 0.8$		$\epsilon = 0.9$		$\epsilon = 1$	
	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w
Max. Temp.	0.0	20.5	0.0	14.4	0.1	11.9	0.2	9.1	0.3	7.2	0.4	5.8	0.5	4.7	0.6	3.6	0.7	2.7	0.8	1.8	0.9	0.9	1.0	0.0
Min. Temp.	0.0	19.7	0.0	13.7	0.1	11.1	0.2	8.1	0.3	6.4	0.4	5.1	0.5	4.0	0.6	3.1	0.7	2.2	0.8	1.4	0.9	0.7	1.0	0.0
Av. Temp.	0.0	17.7	0.0	12.0	0.1	9.4	0.2	7.0	0.3	5.4	0.4	4.4	0.5	3.5	0.6	2.7	0.7	2.0	0.8	1.3	0.9	0.6	1.0	0.0
Wind speed	0.0	33.9	0.0	21.3	0.1	17.2	0.2	12.9	0.3	10.3	0.4	8.3	0.5	6.6	0.6	5.1	0.7	3.6	0.8	2.4	0.9	1.2	1.0	0.0
Precipitation	0.0	107.9	0.1	42.4	0.1	21.2	0.2	9.6	0.3	5.9	0.4	3.9	0.5	2.1	0.6	1.0	0.7	0.3	0.8	0.0	0.8	0.0	1.0	0.0

Table 6: Errors and widths of the prediction regions of CSWF predicting seven days ahead based on sliding window.

	$\epsilon = 0.01$		$\epsilon = 0.05$		$\epsilon = 0.1$		$\epsilon = 0.2$		$\epsilon = 0.3$		$\epsilon = 0.4$		$\epsilon = 0.5$		$\epsilon = 0.6$		$\epsilon = 0.7$		$\epsilon = 0.8$		$\epsilon = 0.9$		$\epsilon = 1$	
	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w	e	w
Max. Temp.	0.0	19.8	0.1	13.7	0.2	11.4	0.3	8.8	0.4	7.0	0.5	5.6	0.6	4.5	0.7	3.5	0.7	2.6	0.8	1.7	0.9	0.9	1.0	0.0
Min. Temp.	0.0	18.6	0.1	13.1	0.2	10.5	0.3	7.9	0.4	6.3	0.5	5.1	0.6	4.0	0.6	3.1	0.7	2.2	0.8	1.4	0.9	0.7	1.0	0.0
Av. Temp.	0.0	17.0	0.1	11.1	0.2	9.1	0.3	6.7	0.4	5.3	0.5	4.3	0.6	3.4	0.7	2.6	0.7	1.9	0.8	1.3	0.9	0.6	1.0	0.0
Wind speed	0.0	33.8	0.0	20.9	0.1	16.8	0.2	12.7	0.3	10.1	0.4	8.1	0.5	6.4	0.6	4.9	0.7	3.5	0.8	2.4	0.9	1.2	1.0	0.0
Precipitation	0.0	102.3	0.1	36.5	0.2	17.9	0.3	8.4	0.4	5.1	0.5	3.1	0.6	1.6	0.7	0.7	0.8	0.1	0.9	0.0	0.9	0.0	1.0	0.0

6.4. Discussion

The experiments clearly show that SWF significantly outperforms the weather forecast domains it combines. As expected, the accuracy of both SWF and the forecasting domains alike is lower for seven-day predictions than it is for next-day predictions. For CSWF the results are less conclusive; although the prediction regions are valid or almost valid, their

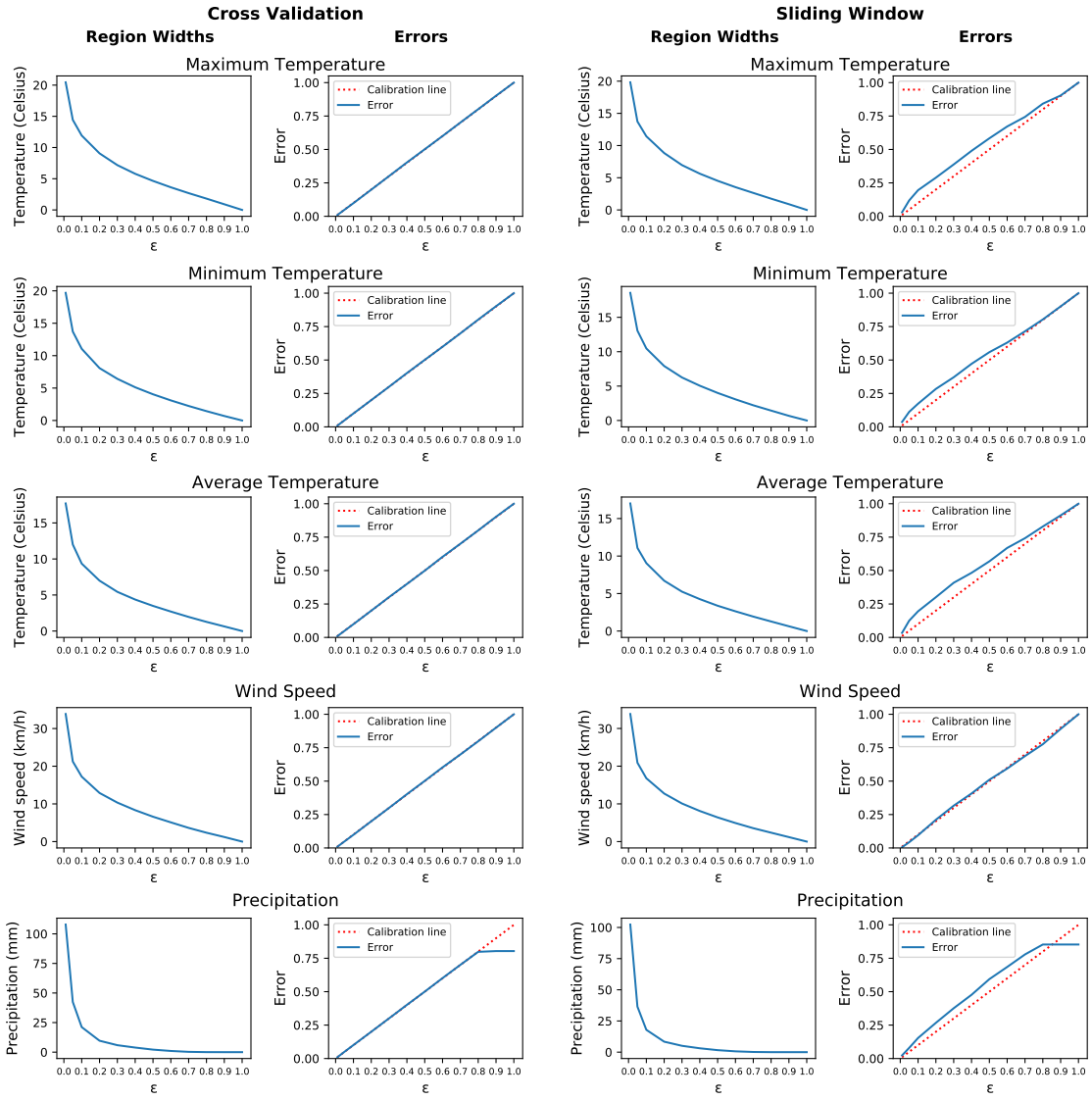

Figure 4: Error and width plots of CSWF predicting seven days ahead based on cross validation (left) and sliding window (right).

width seems relatively high. Since we do not have access to the prediction confidence of the chosen weather forecast domains, we cannot compare the region-width performance and thus cannot derive any conclusions. A potential solution to this problem would be to predict the error of the weather forecast domains based on the gathered data, and using this information to build prediction intervals for these domains. While we would of course have to assume some error, that would allow us to get an indication of the performance of

CSWF compared to the chosen weather forecast domains. As this is not a trivial solution, its potential remains to be explored in future research.

7. Conclusion

In this paper we have introduced the SWF method and CSWF method. They allow us to aggregate the predictions from weather-forecasting domains into more accurate multi-output predictions and to build prediction regions. The methods extend the aggregating procedures of the ensemble methods for weather forecasting beyond averaging by allowing applicability of any type of multi-output regression model. The SWF method and CSWF method do not assume access to the prediction models they aggregate and, thus, demonstrate a general stacking-based approach to boosting the generalization performance of any set of online-prediction services, not limited to weather forecasting.

The experiments with the SWF method and CSWF method have shown the importance of the multi-output predictors. While for the SWF method there is an impressive repertoire of predictors available such as deep neural networks (LeCun et al., 2015), elastic nets (Zou and Hastie, 2005), etc., for the CSWF method there are no conformal multi-output predictors that take into account the output-variable dependencies. This indicates a new research direction in conformal prediction that definitely has to be pursued (in addition to conformal multi-label classification (Papadopoulos, 2014)).

Acknowledgments

We would like to thank Dr. Harris Papadopoulos for providing the Matlab implementation of the conformal nearest-neighbor regression. Furthermore, we would like to thank Markus Dienstknecht, Moritz Haine and Rico Montulet from Maastricht University for their help in gathering the data and their work on the student project from which our research originated.

References

Luai Al Shalabi, Zyad Shaaban, and Basel Kasasbeh. Data mining: A preprocessing engine. *Journal of Computer Science*, 2(9):735–739, 2006.

Peter Bauer, Alan Thorpev, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525:47–55, 2015.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–44, 2015.

Matthew J Menne, Imke Durre, Russell S Vose, Byron E Gleason, and Tamara G Houston. An overview of the global historical climatology network-daily database. *Journal of Atmospheric and Oceanic Technology*, 29(7):897–910, 2012.

Harris Papadopoulos. A cross-conformal predictor for multi-label classification. In *Proceedings of Artificial Intelligence Applications and Innovations - AIAI 2014 Workshops*, volume 437 of *IFIP Advances in Information and Communication Technology*, pages 241–250. Springer, 2014.

Harris Papadopoulos, Vladimir Vovk, and Alexander Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, 40:815–840, 2011.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

David Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

Hailing Zhang and Zhaoxia Pu. Beating the uncertainties: ensemble forecasting and ensemble based data assimilation. *Advances in Meteorology*, 2010:1–10, 2010.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.