Aare Project

Simon Günter

27 12 2019

Aare Project

Introduction/overview/executive

Executive summary

The goal of the project is to create a short-term temperature forecast system for the Aare river in Bern. Two freely available data sources where used to construct the predictor. The best results were achieved with a very simple model.

Proposed prediction task

The Aare is the river flowing through the capital of Switzerland, Bern. It is very popular for swimmer because it has a moderate strong current and the city provides supporting facilities (like changing rooms, swimming pool, space for sunbathing) free of charge. A lot of people working in the city go for a swim after they finished work. The here proposed problem is to predict the water temperature in 4 hours' time from midday to eight o'clock pm.

Sources

All data and source files can be obtained at https://github.com/simonguenter/aare/.

Data

The data sources are the Time series described in https://api.existenz.ch/

- BAFU Hydrology API (Hydro)
- SwissMetNet Weather Measurement API (SMN)

Summary

The key steps of the project were

- Initial choice of locations for which data is collected
- Data collection
- Reducing the feature set
- Data consolidation
- Splitting the data set in training, validation and test set
- Validating different models
- Test the models

Methods/analysis

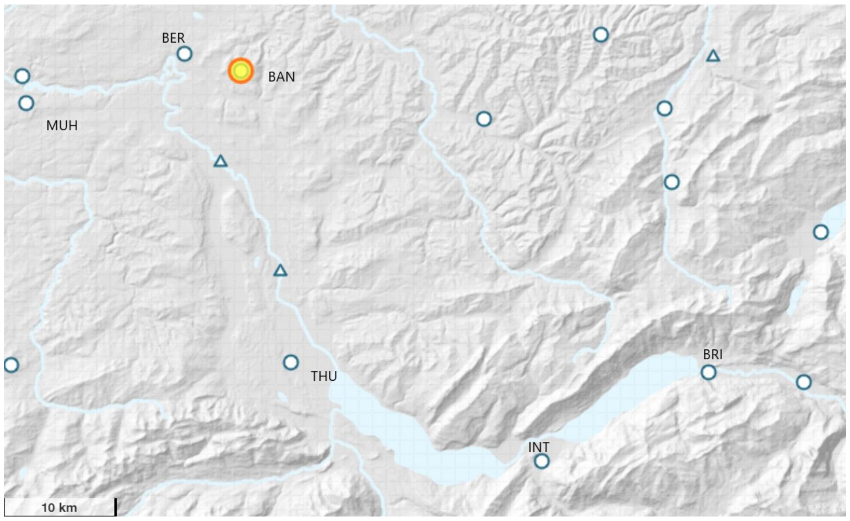
Initial choice of locations

The measurement stations closest to Bern were chosen.

The stations for the SMD were

- Brienz(BRI)
- Interkaken (INT)

- Thun (THU)
- Bantiger (BAN)
- Mühleberg (MUH)
- Bern (BER)



The stations for the Hydro Data were located in

- Brienz (BRI)
- Interkaken (INT)
- Thun (THU)
- Bern (BER)
- Hagneck (HAG)
- Biel (BIE)

Biel/Bienne
Lyss
Hagneck
Burgdorf
Willisau
Ittigen
Bern
Langnau im Emmental
UNESCO Biosphäre Entlebuch
Belp
Schrattenfluh
Naturpark Gantrisch
ibourg
Thun
Brienz
Spiez
Interlaken
Drunengalm

Data collection

The two APIs allow the extraction of data for the last 24 hours. A Java program was written to collect the data for all 12 locations (6 for SME and 6 for hydro). The Java program DataDownloader.java is provided in the GIT repository. A repeated task was set up in Windows to start the Java program every day at the same time. The collection period was from 25.11.2019 to 20.12.2019.

Reducing the feature set

The hydro data has only two features:

- Flow: Speed of Flow of the river
- Temp: Current Temperature of the river

Both are considered relevant to the prediction problem.

The SMN data set has many features and the stations have different feature sets. The features are

- dd Wind direction; ten minutes mean
- dd_tow Wind direction vectoriel; ten minutes interval
- ff Wind speed; ten minutes mean
- ff_tow Wind speed tower; ten minutes mean
- fx Gust peak (one second); maximum
- fx_tow Gust peak (one second) tower; maximum
- qfe Pressure at station level (QFE); current value
- qff Pressure reduced to sea level (QFF); current value
- qnh Pressure reduced to sea level according to standard atmosphere (QNH); current value
- rad Global radiation

- rh Relative air humidity 2 m above ground; current value
- rh_tow Relative air humidity tower; current value
- rr Precipitation; ten minutes total
- ss Sunshine duration; ten minutes total
- td Dew point 2 m above ground; current value
- td_tow Dew point tower
- tt_tow Air temperature
- tt Air temperature 2 m above ground; current value

The following tables show the supported features of the stations

```
##        BRI MUH BAN BER INT THU
## dd       1   0   0   1   1   1
## dd_tow   0   1   1   0   0   0
## ff       1   0   0   1   1   1
## ff_tow   0   1   1   0   0   0
## fx       1   0   0   1   1   1
## fx_tow   0   1   1   0   0   0
## qfe      1   0   0   1   1   1
## qff      0   0   0   1   1   1
## qnh      1   0   0   1   1   1
## rad      0   1   1   1   1   1
## rh       0   0   0   1   1   1
## rh_tow   0   1   1   0   0   0
## rr       1   0   0   1   1   1
## ss       0   1   1   1   1   1
## td       0   0   0   1   1   1
## td_tow   0   1   1   0   0   0
## tt_tow   0   1   1   0   0   0
## tt       0   0   0   1   1   1
```

The follwing features are considered not relevant for the prediction task and are therefore not used: dd,dd_tow,fx,fx_tow, rad. qfe, qff and qnn are measuring the same value with different normalization techniques and are therfore redundant. We will only use qnh.

Consolidating the data

The java Program DataConverter.java, which is provided in the GIT repository, constructs the feature vectors from the file with the raw data. The reason why this step is not done in R is that the raw data is in JSON format and Java has a build-in support for JSON. The constructed feature vectors have the following properties:

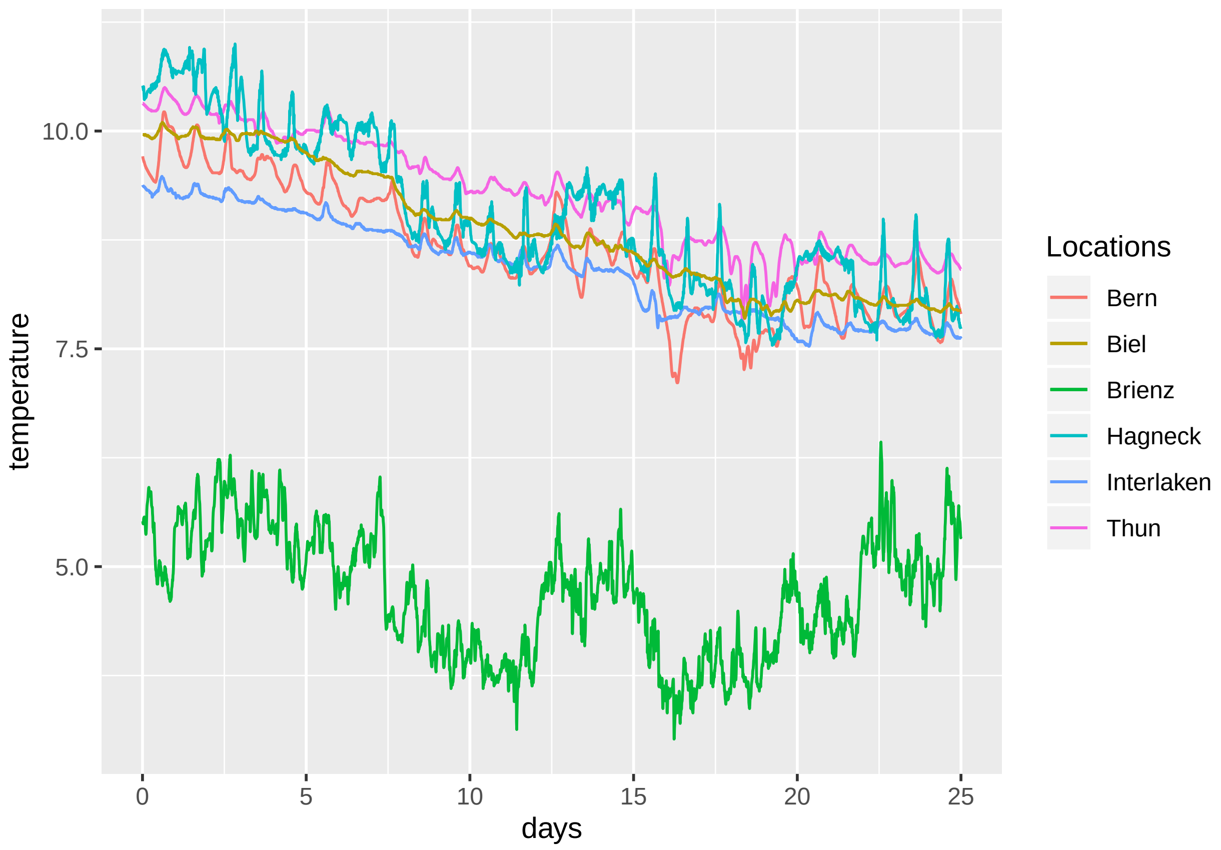
- The feature vector has 46 elements (see table below for numbering of the features).
- Each vector corresponds to a 10-minute time span starting on midnight.
- Measurements are mapped to the corresponding time span. In case of several measurements for the same feature and time span only one is used.
- Vectors of 25 days are included resulting in 3600 vectors in total.

```
##            BRI MUG BAN BER INT THU HAG BIE
## hydo temp  10          1   12  14  16  18
## hydro flow 11          2   13  15  17  19
## ff         34          3   27  20
## ff_tow         42  37
## qnh        35          4   28  21
## rh                     5   29  22
## rh_tow         43  38
## rr         36          6   30  23
## ss            44  39  7   31  24
## td                    8   32  25
## td_tow        45  40
## tt_tow        46  41
## tt                    9   33  26
```

The feature vectors are written in the file aare_data.csv which is also provided in the GIT repository.

Visualizing the data

To verify that the data collection and consolidation worked, a plot of temperature data of the locations was created. The plot contains the day-night variations and the the variations of the different locations are strongly correlated as expected.



Removing missing values

There a total of 1544 missing values in the data set. These values are set to the last previous non missing value of the same feature.

Splitting the data set in training, validation and test set

Data from 25 days were collected. To have a realistic splitting, the split was done on a day basis. As the data set is very small, a 100-fold cross-validation is applied: The days are 100 times randomly mapped to the training, test and validation set where 13 days are used for the training set and 6 days for training and test set, respectively. When reporting details of the models (e.g. coefficients) a fixed non-randomized splitting was used.

Build correspondence of feature set and value to be predicted

The value to be predicted, the water temperature of Bern in 4 hours' time, will be called the goal in the following. In a first step the goal (temperature at 1200 midday to 2000 at night) are matched to the corresponding features (from 800 in the morning to 1600 in the afternoon).

Models

Data science is explorative, so the following sections contain descriptions of the models, their rationales, and their result on the validation sets.

Static model

The simplest model is that the temperature in 4 hours' time is the same as now. On the validation sets a RMSE of 0.2828561 was achieved.

Constant difference model

In this model the difference of the temperature in 4 hours to the actual temperature is constant. This constant is the mean over the whole training set of the difference between the water temperature of Bern and the goal. On the validation sets a RMSE of 0.2344857 was obtained.

Linear model (LM)

In this model goal=c + p1 * f1 + p2 * f2 + … + p46 * f46 with c a constant, f the features and p the parameters. The parameters were determined by linear regression using the function lm. On the validation sets a RMSE of 0.3036546 was achieved. The reason why this RMSE is worse than for the static model is that the linear model is heavily overfit on the training set. This can be confirmed by looking at the coefficients of the linear model of the non-randomized splitting:

```
##            (Intercept)              Bern_flow       Bern_temperature
##          -2.571181e+01          -3.016203e-02          -5.461576e-01
##                 BER_ff                BER_qnh                 BER_rh
##           6.851325e-03           4.618076e-02           7.784456e-03
##                 BER_rr                 BER_ss                 BER_td
##           5.985623e-02           8.137615e-03          -1.554646e-02
##                 BER_tt            brienz_flow     brienz_temperature
##           3.499945e-02           8.629850e-04           1.500716e-01
##        interlaken_flow interlaken_temperature              thun_flow
##           7.508695e-02           4.372511e-01          -8.673632e-03
##       thun_temperature           hagneck_flow    hagneck_temperature
##           3.214506e-02          -4.480771e-04           1.538060e-01
##              biel_flow       biel_temperature                 THU_ff
##           3.484126e-03           4.661735e-01           4.369857e-03
##                THU_qnh                 THU_rh                 THU_rr
##          -1.399368e-02           3.834387e-03          -7.508417e-02
##                 THU_ss                 THU_td                 THU_tt
##          -1.131158e-03          -2.213728e-03           5.303191e-02
##                 INT_ff                INT_qnh                 INT_rh
##           3.188433e-03           3.000329e-02          -2.264663e-02
##                 INT_rr                 INT_ss                 INT_td
##          -7.091183e-02          -2.406212e-03           1.242305e-01
##                 INT_tt                 BRZ_ff                BRZ_qnh
##          -1.077447e-01          -1.410790e-03          -3.223264e-02
##                 BRZ_rr             BAN_ff_tow             BAN_rh_tow
##           2.646296e-02          -1.681227e-03          -4.145866e-04
##               BAN_ss            BAN_td_tow            BAN_tt_tow
##         5.212787e-03          2.984071e-02          5.259651e-02
##           MSK_ff_tow            MSK_rh_tow                MSK_ss
##         5.638041e-03         -1.164990e-02         -1.877735e-03
##           MSK_td_tow            MSK_tt_tow
##         5.846038e-02         -1.037151e-01
```

The coefficients are all over the place. Especially strange is that the coefficient for the feature "temperature Bern", which should be the most strongly correlated to the goal (as it is same feature 4 hours later), has a negative coefficient.

Linear model only Bern features

To avoid overfitting only a subset of all features can be used. An obvious choice is to only select the features of location Bern (which are feature number 1 to 9). On the validation sets a RMSE of 0.25827884 was achieved which is still worse than the RMSE for the constant difference model. For the non-randomized splitting the coefficient of this model for "temperature Bern" is a much more reasonable 0.8762307.

Linear model only Bern Hydro features

To test if better results can be achieved with fewer features, the feature set was further reduced to the hydro measurements of Bern (flow and water temperature). On the validation set a RMSE of 0.2279663 was achieved, the best result at this time.

Hourly models

The difference between the actual water temperature and the temperature in 4 hours strongly depends on the actual time. E.g. from 12:00 to 16:00 a significant increase can be expected and from 16:00 to 20:00 a slight decrease is normal. To take this into account we split the prediction task into a task for each individual daytime hour resulting in 8 tasks. We denote models using such split as "hourly" in the following.

Hourly constant difference model

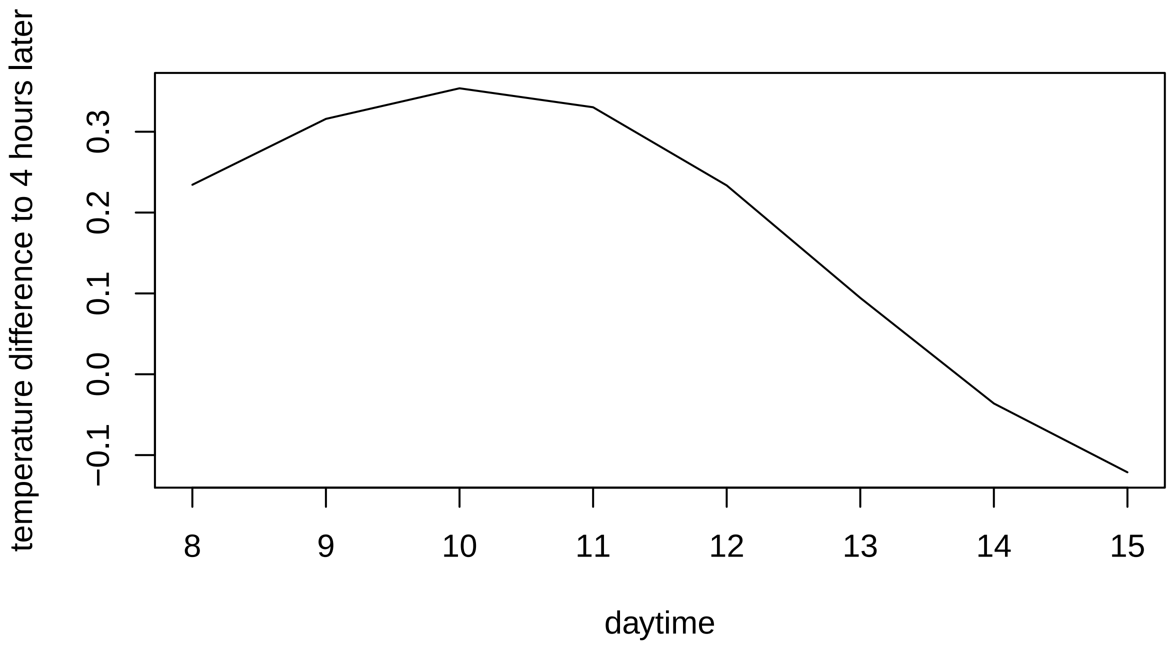
This is the same as the constant difference model but using a different constant for each daytime hour. An excellent RSME of 0.1599026 was achieved on the validation sets.

Hourly linear model only Bern features

This is same as the linear model only using Bern features where a model for each daytime hour is constructed. A RSME of 0.1844289 was obtained on the validation sets.

Linear model only Bern features and Daytime

This linear model also only uses the Bern features. Instead of constructing 8 different models the daytime is added to the feature vector (as number of multitudes of 10 minutes after midnight). A RSME of 0.2068088 was achieved on the validation sets. This much better that the performance of the linear model without the time information but far worse than the hourly model. The reason for this is that the relationship between the time feature and the water temperature is not linear as shown in the following figure for the non-randomized splitting:



Hourly linear model only Bern Hydro features

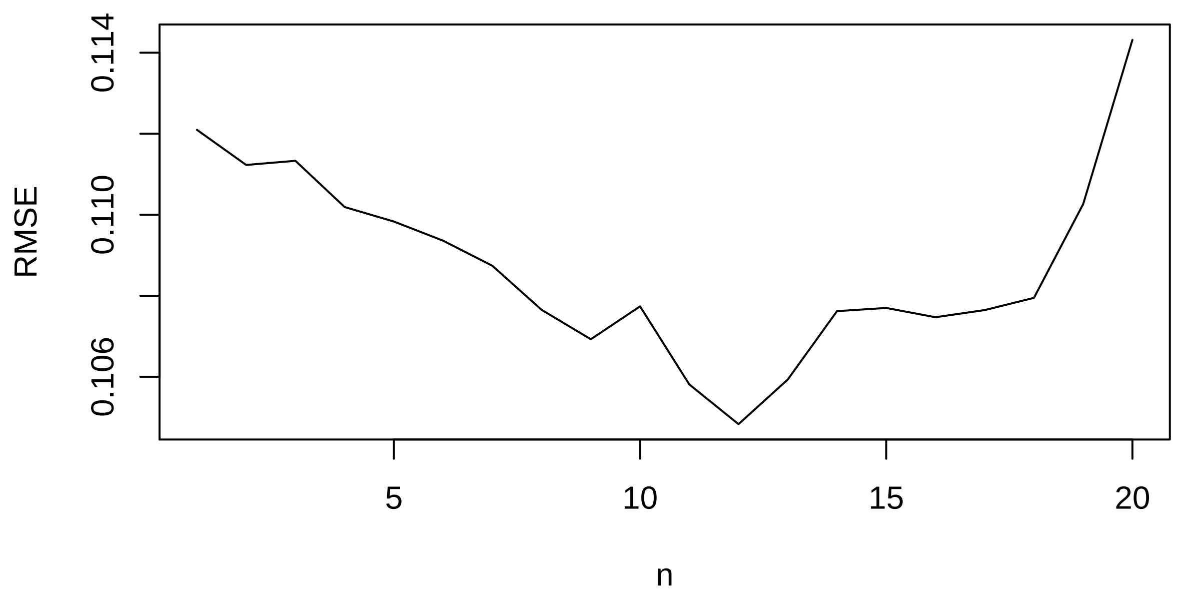
This is the same as the linear model using only Bern Hydro features where a model for each daytime hour is constructed. A RSME of 0.168865 was achieved on the validation sets.

Hourly LM with history using only features from Location Bern.

It may be useful to not only consider the actual measurements but also previous measurements. This model uses as feature vector the current feature values and the feature values 10 minutes ago so that the feature vector has the size of 18. With this model a RSME of 0.1961641 was achieved on the validation sets. This is worse that when only using the original feature vector. A reason for this is again overfitting. This is also indicated by the warning produced by R “prediction from a rank-deficient fit may be misleading”.

Hourly LM with temperature history using only features from Location Bern.

In the last model we again had too many features and overfitting occurred. In this model we only use the the previous values of the most important feature, the water temperature in Bern. To the feature vector the last n values of Bern’s water temperature is added. The optimal value of n is determined on the validation sets. The results on the validation set for the non-randomized splitting and different values of n are given in the following figure



A RSME of 0.1779884 was achieved on the validation sets.

Hourly LM with temperature history using only hydro features from Location Bern.

For this model the current flow and temperature and the last n temperature measurements are considered. The optimal value of n is determined on the validation sets. A RSME of 0.1597465 was achieved, the best result on the validation sets.

Results

The results of the most important models on the validation and test sets are given in the following table

```
##                                             Validation      Test
## Constant difference model                    0.2344857 0.2370119
## LM using only feature Bern                   0.2582788 0.2663168
## Hourly Constant difference model             0.1599026 0.1569507
## Hourly LM features Bern                      0.1844289 0.1817290
## Hourly LM hydro features Bern                0.1688650 0.1662575
## Hourly LM with history features Bern         0.1779884 0.1797779
## Hourly LM with history hydro features Bern  0.1597465 0.1600380
```

The differences between the validation and test set results are very small. This is due to the large number of cross-validations done.

On the test sets the best result was obtained with the "hourly constant difference" model, a very simple model. The best model on the validation set, "Hourly LM with temperature history using only features from Location Bern", is only the second best on the test sets. The reason why this model did better on the validation sets is that the parameter n was optimized on the validation sets, so again overfitting (this time on the validation sets) occurred.

The reason why the simple models outperform the more complex ones is that the data set is very small.

Conclusion / Future work

Several models for a short-term temperature forecast system were constructed and evaluated. Due to the small data set the best performance was achieved with the very simple “hourly constant difference” model.

Future works may include the following:

- Collection of data from a much longer time period. This would reduce the observed overfitting and it would be expected that more complex models outperform simpler ones. In addition, it it is to be expected that with more data the information of the other locations may improve the results.
- Model for other prediction problems, e.g. one day forecast or forecast for a different location.
- Ideally the data would be collected during the summer, as the prediction is most relevant for this season.