1 Quantile Spread Adjustment

The general idea behind the Quantile Spread Adjustment (QSA), is to adjust the spreads of each forecasted quantile by some factor. Quantile spreads are defined as the distance between the respective quantile and some basis. As basis three different points in the forecasting spectrum come into question: the median, the next inner neighbor and the symmetric interval quantile. The quantile spread for the different basis are illustrated in 1.

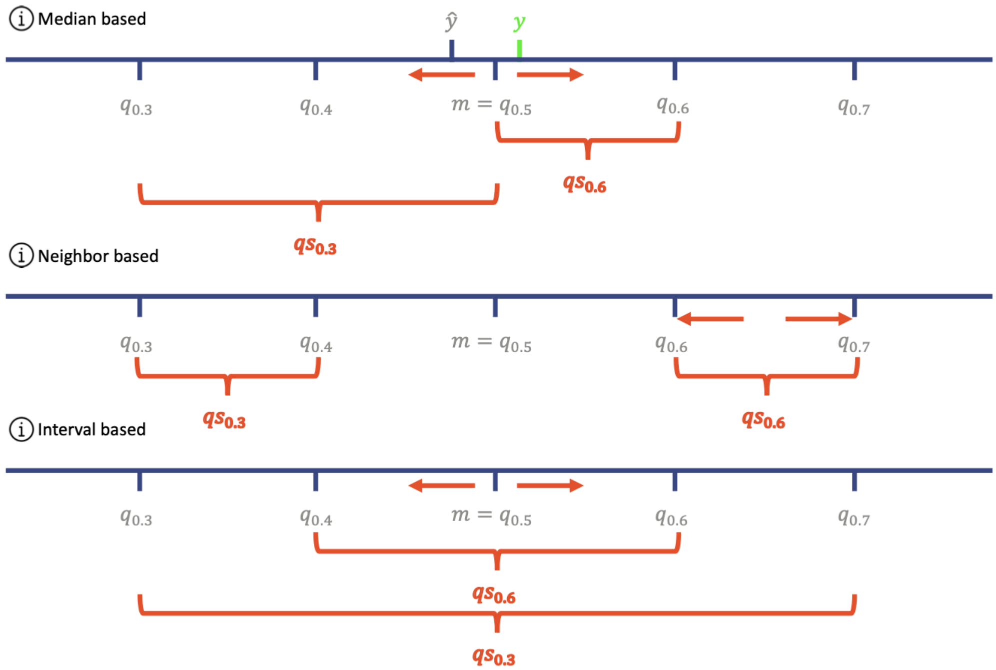

Figure 1: Quantile Spreads for different Basis

We choose the median based definition of the quantile spreads for two main reasons. First, in contrast to the neighborhood based definition, the median basis has the advantage that different quantile spreads are independent of one another. This property makes finding the optimal quantile spread adjustments for a large set of quantiles much simpler. However it comes at the cost that theoretically adjustments can lead to quantile crossing, which would not be the case for neighborhood based adjustments. Our second reason to use the median basis is that it doesn't restrict adjustments to be symmetric for quantile pairs, as would is the case for the interval based approach.

1.1 Theory

Using the median based definition, the next step is to determine how to optimally adjust the quantile spreads. As target function, QSA uses the Weighted Interval Score (reference). Equation (reference), with the number of confidence intervals p, the certainty level of a confidence interval α_p and the number of observations n, shows how the QSA weights $\mathbf{w}$ influence the WIS.

$$
\begin{aligned}
\mathbf{w}^* &= \underset{\mathbf{w}\in\mathbb{R}^p}{\arg\min}\, WIS(\mathbf{y}) \\
&= \underset{\mathbf{w}\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{p} \frac{\alpha_i}{2} \sum_{j=1}^{n} (u^*_{i,j} - l^*_{i,j}) + \frac{2}{\alpha_i} \cdot (l^*_{i,j} - y_j) \cdot \mathbf{1}(y_j \leq l^*_{i,j}) + \frac{2}{\alpha_i} \cdot (y_j - u^*_{i,j}) \cdot \mathbf{1}(y_j \geq u^*_{i,j}) \\
\text{s.t.} \quad & l^*_{i,j} = l_{i,j} + (l_{i,j} - m) \cdot (w^l_i - 1) \quad \text{and} \quad u^*_{i,j} = u_{i,j} + (u_{i,j} - m) \cdot (w^u_i - 1)
\end{aligned}
$$

For a given prediction interval level of α_i, by varying the QSA factor w_i^l for the lower and w_i^u for the upper bound, QSA moves the quantiles from their original values $l_{i,j}$ and $u_{i,j}$ to their adjusted values $l_{i,j}^*$ and $u_{i,j}^*$. QSA factor values larger than 1 lead to an increase in the prediction interval, thus $w_i^l > 1$ reduces the value of $l_{i,j}^*$ and $w_i^u > 1$ increases the value of $u_{i,j}^*$. These changes have two effects, on the one side an increase in w_i^l and w_i^u reduces the sharpness and increases the WIS, on the other side the increased interval may capture more observation which reduces the under- and overprediction penalities in the WIS. Thus depending on the positions of the observed values and predicted quantiles, QSA will either increase or decrease the interval size in order to minimize the WIS.

The `postforecasts` package implements the QSA optimization in three, the weight vector $\mathbf{w}$ restricting, flavors:`qsa_uniform`, `qsa_flexible_symmetric` and `qsa_flexible`. These are listed in equations (reference).

$$
\begin{aligned}
uniform: w_i = c \quad i \in [0, 1, \ldots, p-1, p], \quad c \in \mathbb{R} \\
flexibel\_symmetric: w_i = w_{p-i} \quad c_i \quad i \in [0, 1, \ldots, \frac{p}{2} - 1], \quad c_i \in \mathbb{R} \\
flexibel: w_i \in \mathbb{R}
\end{aligned}
$$

`qsa_uniform` restricts all weight vector values to be identical. `qsa_flexible_symmetric` only restricts pair wise adjustments to be identical. It essentially represents unrestricted QSA with interval based adjustments. Finally `qsa_flexible` is completely unrestricted as each quantile is adjusted separately.

In addition to different flavors, the `postforecasts` package also provides the option to regularize the optimization. Equation (reference) depicts the penalization term that is added to the WIS. It is designed to penalize differences between weight vector values by adding a factor proportional to the sum of squared deviation of the weight vector values from there mean. It therefor regularizes towards the `qsa_uniform` method and only has an effect for the `qsa_flexible_symmetric` and `qsa_flexible` flavors.

$$
\begin{aligned}
&\mathbf{w}^* = \underset{\mathbf{w} \in \mathbb{R}^p}{\arg\min} \; WIS_\alpha(\mathbf{y}) + r \cdot Pen(\mathbf{w}), \quad Pen(\mathbf{w}) = \sum_{i=1}^{p} (w_i - \bar{w})^2 \\
\text{s.t.} \quad &\bar{w} = \frac{1}{p} \sum_{i=1}^{p} w_i
\end{aligned}
$$

1.2 Optimization

Underneath the hood, `postforecasts` accesses the `optim` function from the R package stats[1] package. From the in `optim` available optimization methods, `BFGS` and `L-BFGS-B` turned out to be the most reliable for QSA. `BFGS` is named after Broyden, Fletcher, Goldfarb and Shanno and a quasi-Newton method. `L-BFGS-B`, is a limited memory version of `BFGS` and additionally also support box constraints. As default value we set the optimization method to `L-BFGS-B` as it converges faster than `BFGS` in our data set, due to its limited memory property. The time gain is especially important for the `qsa_flexible_symmetric` and `qsa_flexible` methods which take considerably longer than `qsa_uniform` for a large number of quantiles. Furthermore `L-BFGS-B` also has the advantage that we can lower bound the Quantile Spread factor to not drop below zero, hence we can exclude quantile crossing with the median. The optimization method can be accessed in the function `update_predictions` functions by setting the `optim_method` argument. For `L-BFGS-B`, the lower and upper bound box constraints can be set with the arguements `lower_bound_optim` and `upper_bound_optim`. Besides the use of `optim`, `postforecasts` also provides a line search optimization which is used by passing `line_search`. As the run time increases exponentially with the parameter spaces, this method is currently restricted to the `qsa_uniform`. Here, the method runs QSA for all values of the QSA factor within a sequence. This sequence is defined by its upper and lower values set with the arguments `lower_bound_optim` and `upper_bound_optim` as well as its step size set by `steps_optim`. Regarding the QSA optimization functions shape, there is a potential issue: Due to the trade-off between sharpness and coverage defining the WIS, it can happen that an interval of values for the QSA factor result in the same

[1] https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/optim

score. This becomes less likelier the more observations and quantiles are available, nevertheless it still has to be kept in mind. The `line_search` optimization handles potentially multiple optima by choosing the value closest to 1, hence the smallest possible adjustment of the quantiles. In essence this is a regularization. For the `BFGS` and `L-BFGS-B` this simply means that both methods can converge to different optima while attaining the same WIS. In a future version of the package we aim to tackle this by adding a line search after the use of `BFGS` and `L-BFGS-B` in order to find the optima closest to 1 and thereby regularize the results..