Technische Universität München

Department of Mathematics

Master's Thesis

Estimation of Prediction Intervals for Next-Day Air Temperatures in South Korea Using D-vine Copula Based Quantile Regression

Niklas van Heiss

Supervisor: Prof. Claudia Czado

Advisor: Prof. Claudia Czado

Submission Date: 31.12.2020

I assure the single handed composition of this master's thesis only supported by declared resources.

Garching, December 31, 2020

Acknowledgement

I would like to express my deepest appreciation to Prof. Claudia Czado for the possibility to write my master's thesis under her supervision. I am very grateful for her guidance and helpful advice. Due to Covid restrictions, we frequently arranged online meetings where she always took the time to go through my current progress and discuss open questions. She provided me with new input and encouraged me when things did not work out. Especially in the beginning of my research, the data I selected for my analysis turned out to not suit my purpose. After finding a suitable set of data on the UCI machine learning repository, I could start to successfully work on this new set of data.

I would also like to thank Marija Tepegjozova, who joined our meetings during the last weeks, for her valuable suggestions.

Finally, I wish to wholeheartedly thank my family and friends for their unconditional support and encouragement during all my years of study.

Abstract

In regression analysis, prediction intervals give a range of values that is likely to contain a new observation given the outcome of some predictor variables. A straightforward approach to obtain estimates of such intervals is to predict conditional quantiles of the response variable. In recent years, vine copula-based quantile regression has proven to be a flexible and competitive method in the field of quantile regression. In this thesis, we revisit D-vine copula-based quantile regression following Kraus and Czado (2017) to obtain prediction intervals for next-day air temperatures in Seoul, South Korea, given a set of predictor variables. In particular, we investigate the predictive ability of this method compared to linear quantile regression as a reference method. Here, performance is quantified as a scoring that rewards narrow intervals and adds a penalty if an observation misses the interval. At first, we compare three different approaches to select the training set for next-day predictions. In addition to an intuitive rolling window of successive days prior to the next day, we use a refined rolling window including days from previous years, and also a random selection of days. Overall, the refined rolling window showed best prediction results. Secondly, we investigate an extension of the training horizon for the refined rolling window and a threshold for the number of predictors included in the model. Especially for a larger training horizon, the D-vine quantile regression slightly outperforms the linear quantile regression.

Zusammenfassung

In dieser Masterarbeit betrachten wir die D-vine-copula-basierte Quantilsregression wie sie in Kraus and Czado (2017) vorgestellt wird. Mittels Quantilsregression können beliebige bedingte Quantile einer Zielvariablen in Abhängigkeit von einer Menge von Kovariablen vorhergesagt werden. Unser Ziel ist es, die Vorhersagefähigkeit von D-vine-copula-basierter und linearer Quantilsregression anhand von Vorhersageintervallen für Temperaturwerte zu vergleichen. Ein Vorhersageintervall gibt einen Bereich von Werten an, in dem ein zu vorhersagender Wert mit einer gewissen Wahrscheinlichkeit liegt. Als die Endpunkte solcher Intervalle können wir vorhergesagte bedingte Quantile der Zielvariablen festlegen. Die Güte der vorhergesagten Intervalle bemessen wir mit einem Wert, der sich aus der durchschnittlichen Breite der Intervalle und einem Strafterm für Beobachtungen, die nicht im vorhergesagten Intervall liegen, zusammensetzt. Zu Beginn unserer Auswertung untersuchen wir, wie sich die Auswahl der Trainingsdaten auf die Vorhersagefähigkeit auswirkt. Dabei vergleichen wir drei verschiedene Ansätze: ein aneinanderhängendes Zeitfenster vorausgegangener Tage, ein verfeinertes Zeitfenster, das sich aus vorausgegangenen Tagen des zu vorhersagenden Tages zusammensetzt und einer zufälligen Auswahl vorausgegangener Tage. Insgesamt lieferte das verfeinerte Zeitfenster die besten Vorhersageergebnisse. Im zweiten und dritten Schritt unserer Auswertung vergrößern wir das verfeinerte Zeitfenster und reduzieren die Anzahl der Kovariablen in den Modellen. Insbesondere für größere Zeitfenster beobachten wir leicht bessere Ergebnisse für die D-vine-copula-basierte Quantilsregression im Vergleich zur linearen Quantilsregression.

Contents

List of Figures

List of Tables

1 Introduction

Inferring causal relations from observable quantities has ever since attracted mankind. Beyond doubt, it has induced substantial advancements in astrology, physics and other sciences. From a mathematical perspective, regression analysis can provide appropriate tools. At the core of regression analysis is estimating the relationship between a dependent and one or several independent variables. Exploring such relations serves at least two main purposes: statistical inference and prediction (Sen and Srivastava, 1990). A type of regression analysis that has continually gained importance over the last decades is quantile regression. The concept was introduced by Koenker and Bassett (1978) and aims at estimating conditional quantiles of a response variable given several predictor variables. It finds application in many areas where tail quantiles of conditional distributions are of interest. For instance, Abrevaya (2002) utilize quantile regression techniques to estimate the effects of demographics and maternal behavior during pregnancy particularly at the lower end of the birthweight distribution.

More recently, quantile regression has been enhanced by using vine copula-based approaches as in Aas et al. (2009). Copulas are multivariate distribution functions with uniform margins that establish a relationship between the joint distribution of random variables and their marginal distributions. They serve as building blocks for vine copulas which are multivariate distributions associated with a density construction from bivariate copula and marginal densities. Thus, vine copulas can separate the estimation of the marginals and the dependence structure among the random variables. Further, they allow tail asymmetries and separate multivariate component modelling (Czado, 2019). Especially in finance and economics, copula-based quantile regression has proven to be a flexible and effective analysis and prediction tool. Applications include, for example, financial stress testing (Kraus and Czado, 2017), modelling dependencies in a stock portfolio (Nagler et al., 2019) or between crude oil prices and stock markets (Liao et al., 2019).

In this thesis, we will revisit the D-vine copula-based quantile regression following Kraus and Czado (2017). D-vine copulas are a class of vine copulas that arise from a particular density construction. The motivation of this thesis is a comparative application of D-vine copula-based and linear quantile regression to sizeable real data. More precisely, we aim at estimating prediction intervals for next-day air temperatures. We can readily use predicted conditional quantiles as the lower and upper end of central prediction intervals of the response variable. We will quantify predictive ability by a scoring that rewards narrow intervals and adds a penalty if an observation does not fall in the predicted interval. In our application, we further examine three different training approaches as well as varying training horizons. The data we process originates from Cho et al. (2020) and is available for free download under *https://archive.ics.uci.edu/ml/index.php.* The data set comprises $7,750$ observations of different numerical weather quantities recorded in the urban area of Seoul, capital of South Korea. It includes 21 predictive variables and next-day maximum and minimum air temperatures as responses. With respect to numerical weather data, vine copula-based quantile regression has already been applied in several contexts. For example, Möller et al. (2018) use a D-vine copula-based quantile regression to post-process ensemble temperature forecasts of a weather prediction model showing improvement over benchmark models for larger forecast horizons. Latif and Mustafa (2020) use parametric vine copulas to investigate the critical behavior of flood episodes at a river basin in Malaysia.

All our data processing, model fitting and evaluation is done using `R` software. In particular, we employ the `quantreg` package (Koenker, 2013) to fit linear quantile regression models to our data. For the D-vine copula-based quantile regression we apply the `vinereg()` function from the `vinereg` package (Nagler and Kraus, 2020). For visualization of bivariate copulas we use tools from the `VineCopula` package (Nagler et al., 2020).

Since copulas represent a central element in copula-based quantile regression, we start with a short introduction of the basic concept of copulas before outlining the course of the thesis.

1.1 Concept of Copulas

A copula is a multivariate distribution function. In particular, a d-variate copula C is a multivariate distribution function on the d-dimensional hypercube $[0,1]^d$ with uniformly distributed marginals (Czado, 2019). The corresponding copula density c for an absolutely continuous copula can be obtained by partial differentiation, i.e.

$$c(u_1,\dots,u_d) \coloneqq \frac{\partial^d}{\partial u_1 \dots \partial u_d} C(u_1,\dots,u_d) \text{ for all } \boldsymbol{u} \text{ in } [0,1]^d.$$

Using the copula approach to multivariate data has a key advantage over conventional multivariate data analysis. While we typically consider variables on their original scale and employ pair plots to visualize possible dependence structure between pairs of variables, this brings a critical shortfall. These scatter plots depend on the scale of the variables and therefore the marginal behavior is mixed with the dependence structure of the variables (Czado, 2019). The copula approach will compensate for this by individually modelling the marginals and thus separating marginal effects from the dependence structure of the variables. The marginals are standardized using the probability integral transform:

Definition 1.1 (Probability integral transform (Czado, 2019))**.** Let $X \sim F$ be a continuous random variable and x an observed value of X. The transformation $u \coloneqq F(x)$ is then called the *probability integral transform* (PIT) at x.

Remark 1.2 (Distribution of the probability integral transform (Czado, 2019))**.** The transformed variable $U \coloneqq F(X)$ is uniformly distributed. This can be easily shown:

$$P(U \leq u) = P(F(X) \leq u) = P(X \leq F^{-1}(u)) = F(F^{-1}(u)) = u.$$

Using the probability integral transform we can transform a set of random variables $(X_1,\dots,X_d)$ from original scale (also x-scale) to so-called copula scale (also u-scale):

Definition 1.3 (PIT random variables)**.** For a set of random variables $(X_1,\dots,X_d)$ with marginal distribution functions $X_j \sim F_j$, $j = 1,\dots,d$, we denote the transformed variables

$$U_j \coloneqq F_j(X_j), \quad j = 1,\dots,d$$

as PIT random variables. Then, $(U_1^{(i)}, \dots, U_d^{(i)}) = (F_1(X_1^{(i)}), \dots, F_d(X_d^{(i)}))$, $i = 1, \dots, n$, is called a PIT random sample of size n. If we refer to PIT random variables, we also speak of copula scale or u-scale.

Now, the relationship between multivariate distribution functions and their univariate margins is unveiled by Sklar's Theorem which was first published in Sklar (1959). His theorem is central to the theory of copulas. It shows that we can express multivariate distributions in terms of their marginal distributions and a corresponding copula.

Theorem 1.4 (Sklar's Theorem (Czado, 2019))**.** *Let $\boldsymbol{X}$ be a d-dimensional random vector with joint distribution function F and marginal distribution functions $F_i, i = 1, \dots, d$, then the joint distribution function can be expressed as*

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) \tag{1}$$

with associated density or probability mass function

$$f(x_1, \dots, x_d) = c(F_1(x_1), \dots, F_d(x_d))f_1(x_1) \dots f_d(x_d) \tag{2}$$

for some d-dimensional copula C with copula density c. For absolutely continuous distributions, the copula C is unique. The inverse also holds: the copula corresponding to a multivariate distribution function F with marginal distribution functions F_i for $i = 1, \dots, d$ can be expressed as

$$C(u_1, \dots, u_d) = F(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)) \tag{3}$$

and its copula density or probability mass function is determined by

$$c(u_1, \dots, u_d) = \frac{f(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))}{f_1(F_1^{-1}(u_1)) \cdots f_d(F_d^{-1}(u_d))}. \tag{4}$$

A proof of Sklar's Theorem can be found in Nelson (2006). With Sklar's Theorem, the dependence structure between the standardised variables can now be characterised by copulas. A comprehensive study of copulas is given in Nelson (2006), a more application-oriented guide in Czado (2019). In the course of this thesis we will show that a multivariate density can be constructed from a product of marginal densities and associated bivariate copula densities. The distribution associated with this construction is called a *vine*. Different constructions yield different classes of vines. In D-vine quantile regression, so-called *drawable* vines (D-vines) are utilized. Therefore, we will discuss D-vines in detail. Other classes of vines, such as *regular* and *canonical* vines are studied, amongst others, in Czado (2019).

1.2 Outline

The remainder of the thesis is organized in two main parts. The first part, Chapter 2 and 3, provides the theoretical background of the regression methods that have been applied to the Seoul weather data. In the second part, Chapter 4 and 5, we present and discuss the results of the application study.

The next chapter starts with a short introduction into quantile regression analysis by recalling the method of linear quantile regression. In Chapter 3 we derive the D-vine quantile regression method according to Kraus and Czado (2017) and provide necessary theoretical results from Czado (2019). At the end of Chapter 3 we will give an outline of the D-vine quantile regression algorithm implemented in the `R` package `vinereg`.

In Chapter 4, we briefly depict the characteristics of numerical weather prediction and describe the Seoul weather data. We will then outline the methodology of our application and present the results. Finally, Chapter 5 concludes our findings and gives an outlook on further research.

2 Quantile Regression

This section briefly recalls the idea of linear quantile regression as a natural extension of the ordinary least squares regression method. Astoundingly, it is quantile regression that constitutes the first origins of regression analysis in the history books. Dating back to the mid-18th century, R.J. Boscovich, a Croatian priest, was concerned with the estimation of a median regression slope (Dodge, 2008). Some thirty years later, Laplace adopted Boscovich's ideas to establish an algebraic method. Meanwhile, towards the end of the 18th century, Gauss and Legendre have discovered, without conferring, the least squares principle: fitting a line through the data such that the sum of squared distances is minimized. Based upon their work, linear regression has since evolved into a powerful and widely used tool in data analysis. Mainly due to its computational tractability, the least squares method has happened to outshine quantile regression for many years. It was not until the contributions of Koenker and Bassett (1978) that quantile regression has recurred to continually attract interest in many fields. Finally, the emergence of computers has leveraged quantile regression into a popular counterpart of the least squares method making parameter estimation efficiently feasible. However, both methods differ in an essential point: while ordinary least squares estimates the mean of the response conditioned on the predictor variables, quantile regression can be used to predict any conditional quantile of the response. Through this, it can provide a more complete picture of the stochastic dependencies among the variables. Besides linear quantile regression, there exist numerous other quantile regression methods including local quantile regression (Spokoiny et al., 2013) and semiparametric quantile regression (Noh et al., 2015) which also incorporates a copula-based approach. Though, we will stick to the linear quantile regression as a reference method in our application. Complementary to the ordinary least squares method in mean regression it is one of the most widely known methods in quantile regression. We will now first attend to the ordinary least squares regression, setting up a framework to then discuss the linear quantile regression.

2.1 Ordinary Least Squares Regression

Throughout this thesis, we will denote random variables by capital letters while we use small letters for observed values. In either case, we indicate random vectors and vectors of observed values in bold type. Further, in the regression setting we refer to the response variable as Y and the predictor variables as $\boldsymbol{X} = (X_1, \ldots, X_d)^T$. Corresponding data observations are denoted as $y_i \in \mathbb{R}$ and $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,d})^T \in \mathbb{R}^d$ for $i = 1, \ldots, n$. In the model formulation of the ordinary least squares (abbr. OLS) method we will additionally use the matrix notation

$$\mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{pmatrix} \in \mathbb{R}^{n \times p}$$

for the data, where $p := d+1$. Here, the rows of $\mathbf{X}$ correspond to augmented data vectors $\boldsymbol{x}_i$. We have added a 1 to each vector to account for an intercept in the linear regression model. The notation and derivation generally follows Olive (2017) which gives a detailed review of linear regression. With the matrix notation from above, we can now specify the model which has the following linear form

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_d)^T \in \mathbb{R}^p$ is a set of unknown parameters and $\boldsymbol{\epsilon} \sim N_n(0, \sigma^2 I)$ is a random error term which is assumed to be normally distributed (Groß, 2003). The principle of least squares chooses the parameters such that the expected square loss

$$E[(Y - \boldsymbol{x}^T\boldsymbol{\beta})^2 \mid \boldsymbol{X} = \boldsymbol{x}]$$

is minimized. Minimizing the sample estimate of the square loss, that is the sum of squared differences between the response and its predicted values, leads to the following parameter estimate

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2,$$

where we now used the matrix notation for the data. If we set the first derivative equal to zero, we obtain the closed form solution

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$$

for the parameters of the linear regression model.

If we then use $\boldsymbol{x}_{new}^T\hat{\boldsymbol{\beta}}$ to predict the response, we, in fact, obtain an estimate of the conditional mean of the response variable $E[Y \mid \boldsymbol{X} = \boldsymbol{x}_{new}]$ given the outcome of the predictors (Koenker and Hallock, 2001). This is because the conditional mean is indeed a minimizer of the expected square loss:

$$\underset{h(X)}{\arg\min}\, E[(Y - h(X))^2] = E[Y|X].$$

To show this, we can add and subtract $E[Y|X]$ and resolve the resulting binomial:

$$\begin{aligned}
&E[(Y - h(X))^2] \\
=&E\big[\big((Y - E[Y|X]) - ((h(X) - E[Y|X])\big)^2\big] \\
=&E\big[\big(Y - E[Y|X]\big)^2 + \big(h(X) - E[Y|X]\big)^2 - 2\big(Y - E[Y|X]\big)\big(h(X) - E[Y|X]\big)\big] \qquad (5)
\end{aligned}$$

Using $E\big[E[Y|X]\big] = E[Y|X]|X = E[Y]$ (Law of Total Expectation, (Weiss et al., 2005)), we see that the last term in (5) resolves to 0:

$$\begin{aligned}
E\big[\big(Y - E[Y|X]\big)\big(h(X) - E[Y|X]\big)\big] &= E\Big[E\big[\big(Y - E[Y|X]\big)\big(h(X) - E[Y|X]\big)|X\big]\Big] \\
&= E\Big[\big(\underbrace{E[Y|X] - E[Y|X]}_{=0}\big)\big(h(X) - E[Y|X]\big)\Big]
\end{aligned}$$

Now, we conclude that

$$E[(Y-h(X))^2] = E[(Y-E[Y|X])^2] + \underbrace{E[(h(X)-E[Y|X])^2]}_{\geq 0}$$

$$\geq E[(Y-E[Y|X])^2]$$

with strict equality if and only if $h(X) = E[Y|X]$.

2.2 Linear Quantile Regression

In analogy to the ordinary least squares method, we will now derive the basic idea of parameter estimation in linear quantile regression (abbr. LQR). As preliminaries, we recall the definition of quantiles and the quantile function.

Definition 2.1 (The α-th quantile)**.** For a random variable X with cumulative distribution function F_X, a real number x_α is called the α-th quantile of X if

$$F_X(x_\alpha) \geq \alpha \text{ and } \lim_{t \nearrow x_\alpha} F_X(t) \leq \alpha.$$

If F_X is continuous, the limes coincides with the function value and thus, the α-th quantile simplifies to a solution of

$$F_X(x_\alpha) = \alpha$$

which is not unique if F_X is not strictly monotone. Otherwise the α-th quantile is $x_\alpha = F_X^{-1}(\alpha)$.

To account for the case where a solution would not be unique, we can define the quantile function as a generalized inverse of the cumulative distribution function (abbr. cdf). It chooses the smallest element of possible solutions to be the α-th quantile:

$$F_X^{-1}(\alpha) = \inf\{x : F_X(x) \geq \alpha\} =: q_\alpha(x).$$

In analogy, the conditional quantile function of a response variable Y conditioned on d predictors $\boldsymbol{X} = (X_1, \dots, X_d)^T$ can now be derived from its conditional cdf

$$F_{Y|\boldsymbol{X}}(y \mid x_1, \dots, x_d) = P(Y \leq y \mid X_1 = x_1, \dots, X_d = x_d).$$

For $\alpha \in (0,1)$ the conditional quantile function is then given as (Bernard and Czado, 2015)

$$F_{Y|\boldsymbol{X}}^{-1}(\alpha \mid x_1, \dots, x_d) = \inf\{y \in \mathbb{R} \mid F_{Y|\boldsymbol{X}}(y \mid x_1, \dots, x_d) \geq \alpha\} =: q_\alpha(x_1, \dots, x_d).$$

Like in OLS, where the conditional mean is assumed to be linear in the predictors, we now assume linearity of the conditional quantiles (Handcock and Morris, 1999)

$$q_\alpha(x_1, \dots, x_d) = \boldsymbol{x}^T \boldsymbol{\beta}(\alpha),$$

where $\alpha \in (0,1)$. The unknown parameters are chosen such that the so-called expected *check loss*

$$E[\rho_\alpha(Y - \boldsymbol{x}^T\boldsymbol{\beta}) \mid \boldsymbol{X} = \boldsymbol{x}]$$

is minimized (Yu et al., 2003), where

$$\rho_\alpha(s) := s(\alpha - \mathbb{1}_{s<0})$$

is the *check function.* Taking the sample estimate of the expected check loss leads to the following minimization problem (Hui-Kuang Yu, 2014; Koenker and Bassett, 1978) to determine the unknown parameters:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \sum_{i=1}^{n} \rho_\tau(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}) \\ &= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left[\sum_{i:y_i \geq \boldsymbol{x}_i^T\boldsymbol{\beta}} \alpha|y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}| + \sum_{i:y_i < \boldsymbol{x}_i^T\boldsymbol{\beta}} (1-\alpha)|y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}| \right].\end{aligned}$$

This approach shows strong analogy to the least squares method. Generally, we solve

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \sum_{i=1}^{n} f(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})$$

where $f(\cdot)$ is a loss function (Koenker and Hallock, 2001). In OLS, the quadratic loss function leads to an estimate of the conditional mean, while for LQR the check loss leads to an estimate of conditional quantiles. The latter arises from the fact that the check function asymmetrically weighs the absolute residuals. This can be exemplified, if we consider the check function as a generalisation of the absolute value function. In the case of the conditional median, that is $\alpha = 0.5$, the check function $\rho_{0.5}(\cdot) = 0.5|\cdot|$ corresponds to the absolute value function. This symmetry ensures that the minimization equates the number of positive and negative residuals and thus, yields the sample median (Koenker and Hallock, 2001).

More formally, we can show that the quantile function $q_\alpha(x) = \inf\{x : F_X(x) \geq \alpha\}$ minimizes the expected check loss:

$$\arg\min_{\hat{x}} E[\rho_\alpha(X - \hat{x})].$$

If we rewrite the expected check loss

$$\begin{aligned}E[\rho_\alpha(X-\hat{x})] &= \int_{-\infty}^{\infty} (\alpha(x-\hat{x}) - \mathbb{1}_{x<\hat{x}}(x-\hat{x}))f_X(x)\mathrm{d}x \\ &= \int_{-\infty}^{\hat{x}} (\alpha-1)(x-\hat{x})f_X(x)\mathrm{d}x + \int_{\hat{x}}^{\infty} \alpha(x-\hat{x})f_X(x)\mathrm{d}x,\end{aligned}$$

we can differentiate it with respect to $\hat{x}$. Here, f_X denotes the density function of the random variable X. Under certain regularity conditions and using the formula for differentiation under the integral sign we obtain

$$\begin{aligned}
\frac{\partial}{\partial \hat{x}} E[\rho_\alpha(X-\hat{x})] &= \frac{\partial}{\partial \hat{x}} \int_{-\infty}^{\hat{x}} (\alpha-1)(x-\hat{x}) f_X(x)\mathrm{d}x + \frac{\partial}{\partial \hat{x}} \int_{\hat{x}}^{\infty} \alpha(x-\hat{x}) f_X(x)\mathrm{d}x \\
&= \frac{\partial}{\partial \hat{x}}\hat{x}(\hat{x}-\hat{x}) f_X(\hat{x}) + \int_{-\infty}^{\hat{x}} \frac{\partial}{\partial \hat{x}}\Big((\alpha-1)(x-\hat{x}) f_X(x)\Big)\mathrm{d}x \\
&\quad - \frac{\partial}{\partial \hat{x}}\hat{x}(\hat{x}-\hat{x}) f_X(\hat{x}) + \int_{\hat{x}}^{\infty} \frac{\partial}{\partial \hat{x}}\Big(\alpha(x-\hat{x}) f_X(x)\Big)\mathrm{d}x \\
&= (1-\alpha)\int_{-\infty}^{\hat{x}} f_X(x)\mathrm{d}x - \alpha\int_{\hat{x}}^{\infty} f_X(x)\mathrm{d}x \\
&= (1-\alpha)\int_{-\infty}^{\hat{x}} \mathrm{d}F_X(x) - \alpha\int_{\hat{x}}^{\infty} \mathrm{d}F_X(x) \\
&= (1-\alpha)F_X(\hat{x}) - \alpha(1-F_X(\hat{x})) \\
&= F_X(\hat{x}) - \alpha.
\end{aligned}$$

Now, we see that for $\hat{x} = q_\alpha(x)$ the expected check loss is minimized. In the proof, we followed Tepegjozova (2019). A more involved proof that the conditional quantile function $q_\alpha(x_1, \ldots, x_d)$ indeed minimizes the expected check loss

$$\underset{h(X)}{\arg\min}\, E[\rho_\alpha(Y-h(X))] = q_\alpha(x_1, \ldots, x_d)$$

can be found in Koenker (2005).

Contrary to parameter estimation in OLS, the optimization problem in LQR does not have a closed form solution, but can be solved with standard linear programming techniques, such as the simplex method.

Linear quantile regression has drawn criticism for involving a few drawbacks. Bernard and Czado (2015) point out, that assuming linearity of the conditional quantiles poses a strong assumption that is almost never fulfilled. Furthermore, they criticize linear quantile regression for the possible occurrence of quantile crossing. If the regression lines of different quantile levels have differing slopes, they may cross. At this point, we note that quantile crossing is particularly problematic when using predicted quantiles as upper and lower bounds of prediction intervals. For instance, we can use the predicted conditional 0.025- and 0.975-quantiles to obtain a central 95% prediction interval. If these two quantiles cross, i.e. $\hat{q}_{0.025} > \hat{q}_{0.975}$, we do not obtain a reasonable and interpretable prediction interval. Other typical issues include multicollinearity, necessary variable transformations and interactions between variables that have to be accounted for. D-vine based quantile regression automatically takes care of these issues and allows for a more flexible and less restrictive modelling as will become clearer in the following chapter.

3 D-vine Quantile Regression

We will now discuss the theory of D-vine copulas and outline the method of D-vine copula-based quantile regression (abbr. DVQR). Kraus and Czado (2017) introduce D-vine copula based quantile regression as a semi-parametric method that is designed for continuous input data. It is semi-parametric in the sense that it estimates marginal distribution functions nonparametrically, while the associated copulas are estimated from parametric families. Nagler et al. (2017) provide several methods to also estimate bivariate copulas nonparametrically. In Schallhorn et al. (2017) the D-vine quantile regression approach by Kraus and Czado (2017) has been extended to the case, where some of the input variables are discrete. Panagiotelis et al. (2012) suggest pair copula constructions for entirely discrete input data.

The central idea of D-vine copula based quantile regression is to derive an easily calculable estimate of the conditional quantile function via copulas. The joint density of the input variables can be expressed in terms of a D-vine, that is, a decomposition into a product of bivariate copula densities and marginal densities. It allows to express the conditional quantile function in terms of marginal distribution functions and so-called *h-functions* which are associated with the bivariate copulas in the D-vine. In the course of this section, we will revisit the theory behind the corresponding D-vine quantile regression algorithm implemented in the `R` package `vinereg`. Its implementation is based on Kraus and Czado (2017) and has been extended to allow for nonparametric copulas and mixed discrete-continuous input data. Starting with bivariate copulas, the building blocks of the D-vine, we will continue with the derivation of the D-vine copula based quantile regression model. Where necessary we will provide more tools and results from Czado (2019) to deepen the theory of D-vine copulas.

Instead of modelling the conditional quantile function as a linear function of the predictor variables, we start off with its definition as the inverse conditional distribution function of the response given the predictor variables. Formally, given a response variable Y and some predictor variables $X_1, \ldots, X_d, d \geq 1$ with marginal distribution functions F_Y and $F_j, j = 1, \ldots, d$, respectively, we can denote the conditional quantile function given a quantile level $\alpha \in (0, 1)$ as

$$q_\alpha(x_1, \ldots, x_d) \coloneqq F^{-1}_{Y|X_1,\ldots,X_d}(\alpha \mid x_1, \ldots, x_d).$$

For the remainder of this chapter, we will relate to continuous random variables.

3.1 Bivariate Copulas

As we have mentioned above, bivariate copulas play a key role in the density construction that specifies a D-vine. There are two different estimation approaches of bivariate copulas that are applied in the D-vine regression method: parametric and nonparametric copula estimation.

Before we present bivariate parametric and nonparametric copulas, we introduce a dependence measure to quantify the dependence between two random variables. The *Kendall's*

τ is defined as the probability of concordance minus the probability of discordance of two random variables. It captures the strength and direction of dependence.

Definition 3.1 (Kendall´s tau (Czado, 2019))**.** The Kendall´s τ between the continuous random variables X_1 and X_2 is defined as

$$\tau(X_1, X_2) = P(\underbrace{(X_{11} - X_{21})(X_{12} - X_{22}) > 0}_{\text{concordance}}) - P(\underbrace{(X_{11} - X_{21})(X_{12} - X_{22}) < 0}_{\text{discordance}}),$$

where (X_{11}, X_{12}) and (X_{21}, X_{22}) are independent and identically distributed copies of (X_1, X_2).

Kendall's τ values can range from -1 to 1, where a value of -1 describes perfect negative and a value of 1 perfect positive correlation. The Kendall's τ for two random variables (X_1, X_2) can be estimated from a random sample $\big((x_1, x_2)^{(i)}\big)_{i=1,\dots,n}$ from their joint distribution. We consider all $\binom{n}{2}$ unordered pairs $(\mathbf{x}_i, \mathbf{x}_j) := \big((x_1, x_2)^{(i)}, (x_1, x_2)^{(j)}\big)$, $i, j = 1, \dots, n$, $j \neq i$ and assign it to one of the following four cases:

- *concordant* if either $x_1^{(i)} < x_1^{(j)}$ and $x_2^{(i)} < x_2^{(j)}$ holds or $x_1^{(i)} > x_1^{(j)}$ and $x_2^{(i)} > x_2^{(j)}$. That is, the ordering of the two samples is the same for both variable positions.

- *discordant* if the ordering of the two sample is different at each position, i.e. either $x_1^{(i)} < x_1^{(j)}$ and $x_2^{(i)} > x_2^{(j)}$ holds or $x_1^{(i)} > x_1^{(j)}$ and $x_2^{(i)} < x_2^{(j)}$.

- *extra x_1 pair* if $x_1^{(i)} < x_1^{(j)}$ holds.

- *extra x_2 pair* if $x_2^{(i)} < x_2^{(j)}$ holds.

With the number of occurrences of each of the cases we can now determine an estimate of Kendall's tau.

Definition 3.2 (Estimate of Kendall's τ allowing for ties (Czado, 2019))**.** Let N_c be the number of concordant pairs, N_d be the number of discordant pairs, N_1 be the number of extra x_1 pairs, and N_2 be the number of extra x_2 pairs of a random sample $\big((x_1, x_2)^{(i)}\big)_{i=1,\dots,n}$ of the joint distribution of (X_1, X_2). Then, an estimate of Kendall's τ is given by

$$\hat{\tau}_n := \frac{N_c - N_d}{\sqrt{N_c + N_d + N_1} \times \sqrt{N_c + N_d + N_2}}.$$

The Kendall's τ can be expressed solely in terms of its associated copula. Its value is independent of the marginal distributions. For parametric copulas, the relationship between Kendall's τ and the copula's parameters or generator function can be expressed in closed form (these terms will become clearer in the next section, cf. Equation (6) and (7)). In the context of copulas, the Kendall's τ is therefore a suitable and widely used dependence measure.

In the data application, we will use the empirical Kendall's τ as a first indication of dependence to pre-select eligible predictors for our regression models.

3.1.1 Parametric Copulas

Numerous bivariate parametric copulas have been classified according to their construction (Czado, 2019). As a first class, bivariate *elliptical copulas* emerge from applying the probability integral transform to the margins of known elliptical distributions. We illustrate this approach by constructing the *bivariate Gaussian copula*:

Example 3.3 (Bivariate Gaussian copula (Czado, 2019))**.** The bivariate Gaussian copula can be constructed using a bivariate normal distribution with zero mean, unit variances, and correlation ρ. Applying the inverse statement (3) of Sklar's Theorem we obtain

$$C(u_1, u_2; \rho) = \Phi_2(\Phi^{-1}(u_1), \Phi^{-1}(u_2); \rho),$$

where $\Phi(\cdot)$ is the distribution function of a standard normal $N(0,1)$ distribution and $\Phi(\cdot,\cdot;\rho)$ is the bivariate normal distribution function with zero means, unit variances, and correlation ρ. The corresponding copula density can be expressed as

$$c(u_1, u_2; \rho) = \frac{1}{\phi(x_1)\phi(x_2)} \frac{1}{\sqrt{1-\rho^2}} \exp\left\{ -\frac{\rho^2(x_1^2 + x_2^2) - 2\rho x_1 x_2}{2(1-\rho^2)} \right\},$$

where $x_1 := \Phi^{-1}(u_1)$ and $x_2 := \Phi^{-1}(u_2)$. Here, Equation (4) of Sklar's Theorem has been utilized.

Similarly, the bivariate Student's t copula can be derived as a second member of the class of elliptical copulas. For elliptical copulas, the following relationship between the association parameter ρ and Kendall's τ

$$\rho = \sin\left(\frac{\pi}{2}\tau\right) \tag{6}$$

holds (Czado, 2019). Another class of bivariate copulas is given by so-called *Archimedean copulas* which are constructed from generator functions. For a continuous, strictly monotone decreasing and convex function $\varphi : I \mapsto [0,\infty]$ with $\varphi(1) = 0$, the copula given by

$$C(u_1, u_2) = \varphi^{[-1]}(\varphi(u_1) + \varphi(u_2))$$

is called a bivariate Archimedean copula with generator φ. Here, $\varphi^{[-1]}$ is defined as:

$$\varphi^{[-1]} : [0,\infty] \mapsto [0,1] \text{ with } \varphi^{[-1]}(t) := \begin{cases} \varphi^{-1}(t) & , 0 \le t \le \varphi(0) \\ 0 & , \varphi(0) \le t \le \infty. \end{cases}$$

A member of the class of Archimedean copulas is the *bivariate Clayton copula*:

Example 3.4 (Bivariate Clayton copula (Czado, 2019))**.** The bivariate Clayton copula arises from $\varphi(t) := t^{-\delta} - 1$, $0 < \delta < \infty$ as a generator function:

$$C(u_1, u_2) = (u_1^{-\delta} + u_2^{-\delta} - 1)^{-\frac{1}{\delta}}.$$

The parameter δ controls the degree of dependence. Full dependence is obtained when $\delta \to \infty$. Independence corresponds $\delta \to 0$.

Other well-known members of the Archimedean copula class are the Gumbel, Frank, Joe, BB1, BB6, BB7 and BB8 copula. For Archimedean copulas, we obtain a relationship between the generator function φ and Kendall's τ through the following equation (Czado, 2019)

$$\tau = 1 + 4\int_0^1 \frac{\varphi(t)}{\varphi\prime(t)}\mathrm{d}t. \tag{7}$$

From a third class of copulas, so-called *extreme-value copulas*, we solely mention the *Tawn copula*, as it completes the list of copulas that are employed in the D-vine quantile regression method from the `vinereg` package. It also allows for rotations of copulas that only provide positive dependence ($\tau > 0$). This is the case for the Gumbel, Clayton, Joe, BB1, BB6, BB7, BB8 and Tawn copula. All of these copulas are implemented in the `R`-package `VineCopula`. In Section 3.1.3 a visualization of some copula families will be given.

3.1.2 Nonparametric Copulas

Estimating the bivariate copulas from parametric families bears the risk of misspecification. Most parametric families only allow for highly symmetric and monotone relationships between variables (Nagler et al., 2017) and are unable to capture other relationships. The necessity to estimate pair copulas also nonparametrically is therefore obvious. Nagler et al. (2017) present several approaches to nonparametric copula estimation. While they compare these methods in an extensive simulation study, we will solely discuss kernel estimators. Overall, these methods were found to perform best and are utilized in the D-vine quantile regression method.

Given a random PIT sample $(U_1^{(i)}, U_2^{(i)}) = (F_1(X_1^{(i)}), F_2(X_2^{(i)})), i = 1, \dots, n$, we wish to estimate a corresponding copula density c. Copula density estimation poses the problem of consistency issues at the boundaries of the support (Nagler et al., 2017). As a remedy, the following transformation using the distribution function Φ of a standard normal distribution can be applied:

$$(Z_1^{(i)}, Z_2^{(i)}) = \left(\Phi^{-1}(U_1^{(i)}), \Phi^{-1}(U_2^{(i)})\right).$$

These random vectors now have standard normally distributed margins and their domain is $\mathbb{R}^2$. This is beneficial for kernel density estimators to work well. With Equation (2) from Sklar's Theorem the density of $(Z_1^{(i)}, Z_2^{(i)})$ decomposes to

$$f(z_1, z_2) = c(\Phi(z_1), \Phi(z_2))\phi(z_1)\phi(z_2)$$

for all $(z_1, z_2) \in \mathbb{R}^2$. Here, ϕ denotes the density function of a standard normal distribution. If we set $u_j = \Phi(z_j)$ for $j = 1, 2$, we can rearrange the above equation to

$$c(u_1, u_2) = \frac{f(\Phi^{-1}(u_1), \Phi^{-1}(u_2))}{\phi(\Phi^{-1}(u_1))\phi(\Phi^{-1}(u_2))}\ .$$

Any kernel estimator $\hat{f}$ of f can now be used to define a kernel estimator of the copula density c:

$$\hat{c}(u_1, u_2) = \frac{\hat{f}(\Phi^{-1}(u_1), \Phi^{-1}(u_2))}{\phi(\Phi^{-1}(u_1))\phi(\Phi^{-1}(u_2))}\ .$$

Nagler et al. (2017) propose to estimate $\hat{f}$ by local polynomial likelihood estimators. Therefore, we locally approximate the log-density $\log f(z_1, z_2)$ using the log-quadratic expansion

$$\begin{aligned}\log f(z_1', z_2') &\approx P_{\boldsymbol{a}}(\boldsymbol{z}) \\ &= a_1 + a_2(z_1 - z_1') + a_3(z_2 - z_2') \\ &\quad + a_4(z_1 - z_1')^2 + a_5(z_1 - z_1')(z_2 - z_2') + a_6(z_2 - z_2')^2\end{aligned}$$

for (z_1', z_2') in the neighborhood of $\boldsymbol{z} = (z_1, z_2)$. The coefficients $\boldsymbol{a} = (a_1, \ldots, a_6)^T$ of the polynomial are then identified as the solution to the weighted maximum likelihood problem

$$\begin{aligned}\hat{\boldsymbol{a}} = \underset{\boldsymbol{a} \in \mathbb{R}^6}{\arg\max} \Bigg\{ & \sum_{i=1}^{n} \boldsymbol{K}(B^{-1}\left(\boldsymbol{z} - \boldsymbol{Z}^{(i)}\right)\Big) P_{\boldsymbol{a}}(\boldsymbol{z} - \boldsymbol{Z}^{(i)}) \\ & - n \int_{\mathbb{R}^2} \boldsymbol{K}\left(B^{-1}(\boldsymbol{z} - \boldsymbol{s})\right) \exp\left(P_{\boldsymbol{a}}(\boldsymbol{z} - \boldsymbol{s})\right) ds \Bigg\},\end{aligned}$$

where $\boldsymbol{K}(\boldsymbol{z}) = K(z_1)K(z_2)$ and K is a symmetric probability density function. $B \in \mathbb{R}^{2\times 2}$ is the *bandwidth matrix* with $\det(B) > 0$. Nagler et al. (2017) suggest the following rule of thumb bandwidth matrix for a polynomial order of 2:

$$B_{\text{rot}} = 5n^{-1/10}\hat{\Sigma}_{\boldsymbol{Z}}^{1/2},$$

where $\hat{\Sigma}_{\boldsymbol{Z}}$ is the empirical covariance matrix of $(Z_1^{(i)}, Z_2^{(i)})$, $i = 1, \ldots, n$. An estimate for $f(z_1, z_2)$ is then given by $\exp(\hat{\boldsymbol{a}})$. Finally, the density estimate needs to be normalized to ensure that it actually yields a copula density. This is necessary since the margins may not be uniform in the first place (Nagler, 2016a). The estimator, as shown here, is implemented in the R-package kdecopula (Nagler, 2016b) and utilized for nonparametric copula estimation in the vinereg package.

3.1.3 Visualization

A suitable exploratory visualization tool for bivariate copula-based data analysis is a so-called *normalized bivariate contour plot* (Czado, 2019). Therefore, we consider a new normalized variable scale, the **z-scale**:

$$(Z_i, Z_j) = (\Phi^{-1}(U_i), \Phi^{-1}(U_j)) = ((\Phi^{-1}(F_i(X_i)), \Phi^{-1}(F_j(X_j))).$$

Then, (Z_i, Z_j) features standard normal margins $Z_i, Z, j \sim N(0, 1)$ and density

$$g(z_i, z_j) = c(\Phi(z_i), \Phi(z_j))\phi(z_i)\phi(z_j).$$

Here, $\Phi(\cdot)$ and $\phi(\cdot)$ denote the distribution and density function of a $N(0, 1)$ distributed random variable. Table 1 gives a summary of the different variable scales and their

Variable scale	Transformation	Density
x-scale (*original scale*)	(X_1, X_2)	$f(x_1, x_2)$
u-scale (*copula scale*)	$(U_1, U_2) = (F_1(X_1), F_2(X_2))$	copula density $c(u_1, u_2)$
z-scale (*marginal normalized scale*)	$(Z_1, Z_2) = (\Phi^{-1}(U_1), \Phi^{-1}(U_2))$ $= ((\Phi^{-1}(F_1(X_1)), \Phi^{-1}(F_2(X_2)))$	$g(z_1, z_2)$ $= c(\Phi(z_1), \Phi(z_2))\phi(z_1)\phi(z_2)$

Table 1: Variable scales and their associated densities (Czado, 2019)

associated densities. In a normalized bivariate contour plot, contours of $g(z_1, z_2)$, i.e. $g(z_1, z_2) = k$ for different values of k, are displayed. This transformation to z-scale decisively enhances the interpretability of the contour plots. For example, a Gaussian copula would be recognized as elliptical contours with a center at $(0, 0)$ in the normalized contour plot. Other parametric copula families also assume characteristic shapes in the normalized contour plots.
In Figure 1 we display exemplary contour plots of different parametric copulas. For each copula, the top row of the plot shows the normalized bivariate copula contours (on the z-scale) and in the bottom row corresponding random samples on the copula scale (u-scale) are displayed.

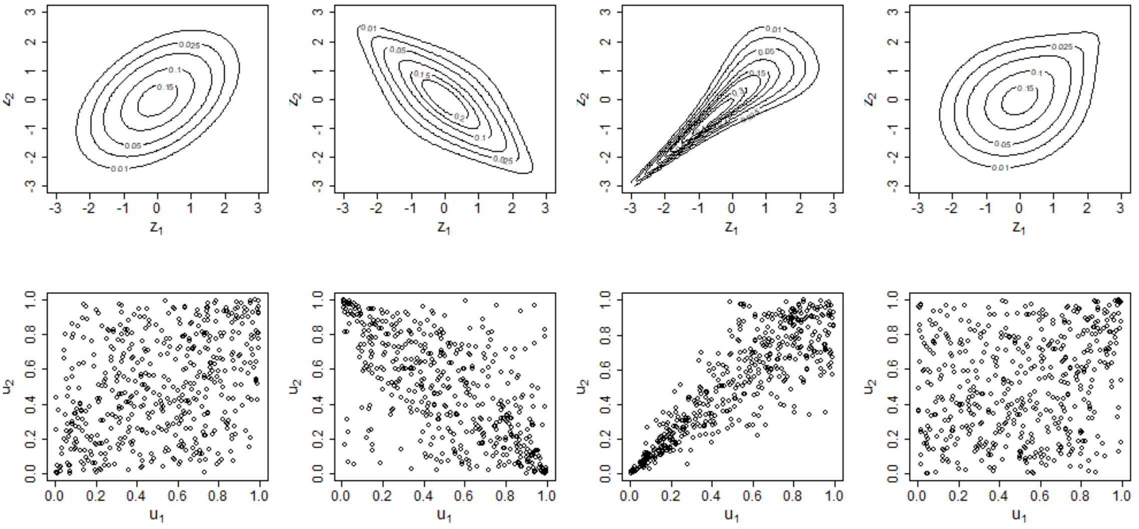


Figure 1: First column: Gaussian copula with $\tau = 0.3$, second column: Student's t copula with $\tau = -0.5$ and $\nu = 4$, third column: Clayton copula with $\tau = 0.7$ and fourth column: Gumbel copula with $\tau = 0.2$. Top row: Normalized bivariate copula contours, bottom row: pairs plot of random samples (u_1, u_2) on the copula scale.

In practical data applications, *empirical normalized contour plots* can be compared to the contour shapes of the parametric copula families to give a first visual indication for an appropriate family selection. Therefore, the data sample needs to be transformed to copula scale and normalized scale. Since the true marginals are usually unknown, the marginal distribution functions have to be estimated first. We can then transform a data sample into so-called *pseudo-copula data*. In Section 3.5.1, we discuss how the marginal densities are estimated in the D-vine regression setting.

Definition 3.5 (Pseudo-copula data)**.** For an independent and identically distributed (i.i.d.) data sample $y^{(i)}, x_j^{(i)}$, $j = 1, \dots, d$ and $i = 1, \dots, n$ and estimated marginals $\hat{F}_Y$ and $\hat{F}_j$, $j = 1, \dots, d$, the pseudo-copula data is given by the PIT values

$$\hat{F}_Y(y^{(i)}) = \hat{v}^{(i)} \text{ and}$$

$$\hat{F}_j(x_j^{(i)}) = \hat{u}_j^{(i)}.$$

We can now transform the pseudo-copula data to the z-scale and estimate corresponding contours through bivariate kernel density smoothing (Czado, 2019). In Section 3.5.2, we describe how the bivariate copulas are estimated in the D-vine quantile regression method.

3.1.4 *h*-functions Associated with Bivariate Copulas

As Sklar's Theorem (see Theorem 1.4) shows, we can express a multivariate distribution function in terms of its marginal distributions and a corresponding copula. The following Lemma links conditional densities and distribution functions with their marginals and an associated bivariate copula for $d = 2$, respectively.

Lemma 3.6. (Czado, 2019) The conditional density and distribution function can be rewritten as

$$\begin{aligned} f_{1|2}(x_1 \mid x_2) &= c_{12}(F_1(x_1), F_2(x_2)) f_1(x_1) \\ F_{1|2}(x_1 \mid x_2) &= \frac{\partial}{\partial u_2} C_{12}(F_1(x_1), u_2)|_{u_2 = F_2(x_2)} \\ &=: \frac{\partial}{\partial F_2(x_2)} C_{12}(F_1(x_1), F_2(x_2)). \end{aligned}$$

Proof. Using Sklar's Theorem and the definition of a conditional density we obtain

$$\begin{aligned} f_{1|2}(x_1 \mid x_2) &= \frac{f_{12}(x_1, x_2)}{f_2(x_2)} \\ &= \frac{c_{12}(F_1(x_1), F_2(x_2)) f_1(x_1) f_2(x_2)}{f_2(x_2)} \\ &= c_{12}(F_1(x_1), F_2(x_2)) f_1(x_1) \\ &= \frac{\partial^2 C_{12}(u_1, u_2)}{\partial u_1 \partial u_2}|_{u_1 = F_1(x_1), u_2 = F_2(x_2)} \frac{\partial u_1}{\partial x_1} \\ &= \frac{\partial}{\partial u_2} \left(\frac{\partial}{\partial x_1} C_{12}(F_1(x_1), u_2) \right) |_{u_2 = F_2(x_2)}. \end{aligned}$$

Using this expression for $f_{1|2}(x_1 \mid x_2)$, we can now proof the second part of the Lemma:

$$\begin{aligned} F_{1|2}(x_1 \mid x_2) &= \int_{-\infty}^{x_1} \frac{\partial}{\partial u_2} \left(\frac{\partial}{\partial z_1} C_{12}(F_1(z_1), u_2) \right) |_{u_2 = F_2(x_2)} \mathrm{d} z_1 \\ &= \frac{\partial}{\partial u_2} \left(\int_{-\infty}^{x_1} \frac{\partial}{\partial z_1} C_{12}(F_1(z_1), u_2) \mathrm{d} z_1 \right) |_{u_2 = F_2(x_2)} \\ &= \frac{\partial}{\partial u_2} C_{12}(F_1(x_1), u_2)|_{u_2 = F_2(x_2)}. \end{aligned}$$

□

Applying Lemma 3.6 to the bivariate copula distribution C_{12} yields

$$C_{1|2}(u_1 \mid u_2) = \frac{\partial}{\partial u_2} C_{12}(u_1, u_2) \ \forall u_1 \in [0,1]. \tag{8}$$

Through this, we obtain a link between the conditional distribution function $F_{1|2}$ and the conditional copula $C_{1|2}$

$$\begin{aligned} F_{1|2}(x_1 \mid x_2) &\overset{\text{Lemma 3.6}}{=} \frac{\partial}{\partial u_2} C_{12}(F_1(x_1), u_2)|_{u_2 = F_2(x_2)} \\ &\overset{(8)}{=} \quad C_{1|2}(F_1(x_1) \mid F_2(x_2)). \end{aligned} \tag{9}$$

Following the inversion rule for a composite function $f \circ g$,

$$(f \circ g)^{-1} = g^{-1} \circ f^{-1},$$

we obtain

$$F_{1|2}^{-1}(u_1 \mid x_2) = F_1^{-1}(C_{1|2}^{-1}(u_1 \mid F_2(x_2))). \tag{10}$$

The conditional distribution function $C_{1|2}$ in Equation (8) is also denoted as *h-function*:

Definition 3.7 (*h*-functions of bivariate copulas (Czado, 2019))**.** The h-functions corresponding to a bivariate copula C_{12} are defined for all $(u_1, u_2) \in [0,1]^2$ as

$$h_{1|2}(u_1 \mid u_2) := \frac{\partial}{\partial u_2} C_{12}(u_1, u_2),$$

$$h_{2|1}(u_2 \mid u_1) := \frac{\partial}{\partial u_1} C_{12}(u_1, u_2).$$

In the following, we will use the relationship in Lemma 3.6 to express the joint density f of a random vector $\boldsymbol{X} = (X_1, \dots, X_d)^T$ in terms of bivariate copula densities and its marginal densities (Kraus and Czado, 2017). Since this approach uses bivariate copula densities, it is also denoted as *Pair-Copula Construction* (PPC), the resulting density is called a *D-vine density* with order $X_1 - X_2 - \cdots - X_d$ (Kraus and Czado, 2017; Czado, 2019). In case all margins are uniform, we speak of a *D-vine copula*.

3.2 Pair-Copula Construction of a D-vine Density

We can now derive the D-vine density that is constructed from bivariate copulas which we also refer to as pair-copulas. We adopt the following notations from (Czado, 2019; Kraus and Czado, 2017) to differentiate between copulas associated with bivariate conditional distribution functions and bivariate conditional distributions on the copula scale, i.e. the distribution of the PIT random variables.

Definition 3.8. Let $(X_1, \dots, X_d)$ be a set of random variables and $(U_1, \dots, U_d)$ the corresponding PIT random variables. For a set of indices $D \subset \{1, \dots, d\}$ and $i, j \in \{1, \dots, d\} \setminus D$,

- $C_{ij;D}(\cdot, \cdot; \boldsymbol{x}_D)$ denotes the copula associated with the bivariate conditional distribution (X_i, X_j) given $\boldsymbol{X}_D = \boldsymbol{x}_D$. The corresponding copula density is denoted by $c_{ij;D}(\cdot, \cdot; \boldsymbol{x}_D)$.
- the conditional distribution function of the PIT random variables (U_i, U_j) given $\boldsymbol{U}_D = \boldsymbol{u}_D$ is expressed as $C_{ij|D}(\cdot, \cdot; \boldsymbol{u}_D)$ with bivariate density function $c_{ij|D}(\cdot, \cdot; \boldsymbol{u}_D)$.
- $F_{i|D}(\cdot, \cdot; \boldsymbol{x}_D)$ denotes the conditional distribution function of the random variable X_i given $\boldsymbol{X}_D = \boldsymbol{x}_D$.

Finally, we use the following abbreviation to shorten notation

$$c_{i,j;D} := c_{i,j;D}(F_{i|D}(x_i \mid \boldsymbol{x}_D), F_{j|D}(x_j \mid \boldsymbol{x}_D); \boldsymbol{x}_D). \tag{11}$$

A D-vine density is obtained through the following density decomposition:

Theorem 3.9 (Drawable vine (D-vine) density (Czado, 2019))**.** *Every joint density $f_{1,\dots,d}$ can be decomposed as*

$$f_{1,\dots,d}(x_1, \dots, x_d) = \left[\prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{i,(i+j);(i+1),\dots,(i+j-1)}\right] \cdot \left[\prod_{k=1}^{d} f_k(x_k)\right], \tag{12}$$

where we used the abbreviation given in (11). The distribution associated with this density decomposition is called a drawable vine (D-vine).

Proof. We start with decomposing the joint density into products of conditional and marginal densities:

$$\begin{aligned} f_{1,\dots,d}(x_1, \dots, x_d) &= f_{d|1,\dots,d-1}(x_d \mid x_1, \dots, x_{d-1}) f_{1,\dots,d-1}(x_1, \dots, x_{d-1}) \\ &= \dots = \left[\prod_{t=2}^{d} f_{t|1,\dots,t-1}(x_t \mid x_1, \dots, x_{t-1})\right] \times f_1(x_1). \end{aligned} \tag{13}$$

Then, applying Lemma 3.6 recursively to the conditional densities in (13) yields the following:

$$\begin{aligned} f_{t|1,\dots,t-1}(x_t \mid x_1, \dots, x_{t-1}) &= c_{1,t|2,\dots,t-1} \cdot f_{t|2,\dots,t-1}(x_t \mid x_2, \dots, x_{t-1}) \\ &= c_{1,t|2,\dots,t-1} \cdot c_{2,t|3,\dots,t-1} \cdot f_{t|3,\dots,t-1}(x_t \mid x_3, \dots, x_{t-1}) \\ &= \dots = \left[\prod_{s=1}^{t-2} c_{s,t;s+1,\dots,t-1}\right] \cdot c_{t-1,t} \cdot f_t(x_t). \end{aligned}$$

Inserting this decomposition into (13) completes the proof:

$$f_{1,\dots,d}(x_1,\dots,x_d) = \left[\prod_{t=2}^{d}\prod_{s=1}^{t-2} c_{s,t;s+1,\dots,t-1}\right] \cdot \left[\prod_{t=2}^{d} c_{t-1,t}\right] \cdot \left[\prod_{k=1}^{d} f_k(x_k)\right]$$

$$\overset{s=i,t=i+j}{=} \left[\prod_{j=1}^{d-1}\prod_{i=1}^{d-j} c_{i,(i+j);(i+1),\dots,(i+j-1)}\right] \cdot \left[\prod_{k=1}^{d} f_k(x_k)\right].$$

□

In DVQR, it is commonly assumed that the copulas associated with conditional distributions do not depend on the values of $\boldsymbol{x}_D$. This assumption is referred to as *simplifying assumption* (Kraus and Czado, 2017):

Definition 3.10 (Simplifying assumption (Czado, 2019))**.** If in the context of Theorem 3.9 the following

$$c_{i,j;D}(F_{i|D}(x_i \mid \boldsymbol{x}_D), F_{j|D}(x_j \mid \boldsymbol{x}_D); \boldsymbol{x}_D) = c_{i,j;D}(F_{i|D}(x_i \mid \boldsymbol{x}_D), F_{j|D}(x_j \mid \boldsymbol{x}_D))$$

holds for all $\boldsymbol{x}_D$ and i, j and D, the corresponding D-vine distribution is called simplified.

In order to visualize the concept of the pair-copula construction, Kraus and Czado (2017) have adopted the graph theoretic representation of a D-vine density which was first established by Bedford and Cooke (2002). For d variables, a D-vine density can be represented as a nested set of $d-1$ trees, where edges in the first tree represent pairwise dependence and edges in subsequent trees represent conditional dependence (Gneiting and Raftery, 2007). Example 3.11 demonstrates this correspondence between pair copulas in the D-vine density and edges in the nested set of trees. The sequence in which the variables appear in the first tree of the D-vine is referred to as the *order* of the D-vine.

Example 3.11. The following density decomposition shows an exemplary 4-dimensional D-vine density with order $X_1{-}X_2{-}X_3{-}X_4$. In Figure 2, its graph theoretic representation is illustrated. Each edge of the nested set of trees corresponds to a pair-copula in the D-vine density.

$$\begin{aligned} f(x_1,x_2,x_3,x_4) =& f_1(x_1)f_2(x_2)f_3(x_3)f_4(x_4) \\ &\cdot c_{12}\big(F_1(x_1),F_2(x_2)\big)\cdot c_{23}\big(F_2(x_2),F_3(x_3)\big)\cdot c_{34}\big(F_3(x_3),F_4(x_4)\big) \quad \text{(T1)} \\ &\cdot c_{13;2}\big(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2)\big)\cdot c_{24;3}\big(F_{2|3}(x_2|x_3),F_{4|3}(x_4|x_3)\big) \quad \text{(T2)} \\ &\cdot c_{14;23}\big(F_{1|23}(x_1|x_2,x_3),F_{4|23}(x_4|x_2,x_3)\big) \quad \text{(T3)} \end{aligned}$$

3.3 Estimate of the Conditional Quantile Function

Back to our initial problem of estimating the conditional quantile function

$$F^{-1}_{Y|X_1,\dots,X_d}(\alpha \mid x_1,\dots,x_d) \tag{14}$$

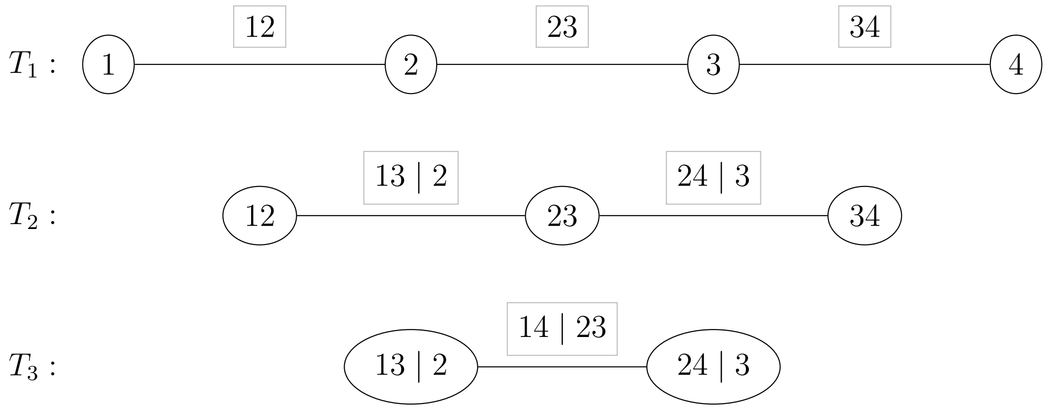


Figure 2: Graph theoretic representation of a 4-dimensional D-vine

via copulas, we can now rewrite (14) in terms of its associated copula. If we use the PIT variables $V := F_Y(Y)$ and $U_j := F_j(X_j)$, $j = 1, \ldots, d$, we can express the conditional distribution function on the original scale as a conditional distribution function on the copula scale (cf. Definition 1.1 and 1.3). With PIT values $v := F_Y(y)$ and $u_j := F_j(x_j)$ for all $j = 1, \ldots, d$ we obtain the following:

$$\begin{aligned} F_{Y|X_1,\ldots,X_d}(y \mid x_1, \ldots, x_d) &= P(Y \leq y \mid X_1 = x_1, \ldots, X_d = x_d) \\ &= P(F_Y(Y) \leq v \mid F_1(X_1) = u_1, \ldots, F_d(X_d) = u_d) \\ &= C_{V|U_1,\ldots,U_d}(v \mid u_1, \ldots, u_d). \end{aligned}$$

Inversion yields the following expression for the conditional quantile function:

$$F^{-1}_{Y|X_1,\ldots,X_d}(\alpha \mid x_1, \ldots, x_d) = F_Y^{-1}\left(C^{-1}_{V|U_1,\ldots,U_d}(\alpha \mid u_1, \ldots, u_d)\right).$$

That is, we can express the conditional quantile function in terms of the inverse marginal distribution function F_Y^{-1} and the conditional copula quantile function $C^{-1}_{V|U_1,\ldots,U_d}$. An estimate of the conditional quantile function can therefore be obtained through estimating the marginals F_Y and F_j for $j = 1, \ldots, d$ as well as the conditional distribution function $C_{V|U_1,\ldots,U_d}$:

$$\hat{q}_\alpha(x_1, \ldots, x_d) := \hat{F}_Y^{-1}\left(\hat{C}^{-1}_{V|U_1,\ldots,U_d}(\alpha \mid \hat{u}_1, \ldots, \hat{u}_d)\right),$$

where $\hat{u}_j := \hat{F}_j(x_j)$ is the estimated PIT value of x_j, for $j = 1, \ldots, d$ respectively. What remains in question is how the conditional distribution $C_{V|U_1,\ldots,U_d}$ can be estimated. The following section will show how this can be achieved.

3.4 Evaluating the Conditional Copula Quantile Function

Kraus and Czado (2017) suggest to fit a D-vine copula to $(V, U_1, \ldots, U_d)^T$. In Section 3.5 we enlarge on how the D-vine copula is sequentially fitted to the data. Through this, we obtain pair copulas from the PCC associated with bivariate conditional distribution functions of the data. Yet, the values of the pair-copulas are not easily calculable since they still have conditional distributions $F_{i|D}(x_i \mid \boldsymbol{x}_D)$ as their arguments. To overcome this, Kraus and Czado (2017) suggest to recursively apply the following formula by Joe (1997) which, for $l \in D$ and $D_{-l} := D \setminus \{l\}$, links the argument of a pair-copula to the h-functions associated with pair-copulas from lower trees:

$$F_{i|D}(x_i \mid \boldsymbol{x}_D) = h_{i|l;D_{-l}}(F_{i|D_{-l}}(x_i \mid \boldsymbol{x}_{D_{-l}}) \mid F_{l|D_{-l}}(x_l \mid \boldsymbol{x}_{D_{-l}})). \tag{15}$$

Having fitted a D-vine to the data, we can then use the recursion to express the conditional copula quantile function in terms of inverse h-functions. The following example will demonstrate this.

Example 3.12. Let us assume we have fitted a D-vine with order $V - U_1 - U_2 - U_3$ to the data. Then, the conditional distribution function of V given $(U_1, U_2, U_3)^T$ can be expressed as

$$\begin{aligned}
C_{V|U_1,U_2,U_3}(v \mid u_1, u_2, u_3) &= h_{V|U_3;U_1,U_2}(C_{V|U_1,U_2}(v \mid u_1, u_2) \mid C_{U_3|U_1,U_2}(u_3 \mid u_1, u_2)) \\
&= h_{V|U_3;U_1,U_2}(h_{V|U_2;U_1}(C_{V|U_1}(v \mid u_1) \mid C_{U_2|U_1}(u_2 \mid u_1)) \mid \\
&\qquad h_{U_3|U_1;U_2}(C_{U_3|U_2}(u_3 \mid u_2) \mid C_{U_1|U_2}(u_1 \mid u_2))) \\
&= h_{V|U_3;U_1,U_2}(h_{V|U_2;U_1}(h_{V|U_1}(v \mid u_1) \mid h_{U_2|U_1}(u_2 \mid u_1)) \mid \\
&\qquad h_{U_3|U_1;U_2}(h_{U_3|U_2}(u_3 \mid u_2) \mid h_{U_1|U_2}(u_1 \mid u_2))),
\end{aligned}$$

where the h-functions are associated with the bivariate pair-copulas of the D-vine.

Eventually, we arrive at the conditional copula quantile function through inversion:

$$\begin{aligned}
&C^{-1}_{V|U_1,U_2,U_3}(\alpha | u_1, u_2, u_3) = \\
&h^{-1}_{V|U_1}\left[h^{-1}_{V|U_2;U_1}\left\{ h^{-1}_{V|U_3;U_1,U_2}\left(\alpha \middle| h_{U_3|U_1;U_2}\left(h_{U_3|U_2}(u_3|u_2) | h_{U_1|U_2}(u_1|u_2) \right) \right) \middle| h_{U_2|U_1}(u_2|u_1) \right\} \middle| u_1 \right].
\end{aligned} \tag{16}$$

Extending these calculations to the case of d predictor variables, the determination of the conditional quantile function $F_Y^{-1}\left(C^{-1}_{V|U_1,\ldots,U_d}(\alpha \mid u_1, \ldots, u_d) \right)$ is easily feasible.

3.5 Outline of the DVQR Algorithm

We have now derived all necessary ingredients to outline the DVQR algorithm that is implemented in the `vinereg()` method. The method is built up of two main estimation processes:

1. Estimate the marginal distribution functions in order to transform input data from original scale to u-scale data (pseudo-copula data)

2. Fit a D-vine copula to the pseudo-copula data

In particular, the second task implies the estimation of bivariate pair-copulas. Having fitted a D-vine copula to the data, we have seen above how we can arrive at an estimate of the conditional quantile function through the h-functions of associated pair-copulas from the pair-copula construction of the vine.

3.5.1 Estimating the Marginal Distributions

In the first step, all marginal distribution functions of the input variables are estimated. Hence, we can transform the input data to pseudo-copula data (cf. Definition 3.5). The `vinereg`-method estimates the marginals nonparametrically (Kraus and Czado, 2017) via the following continuous kernel smoothing estimator

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} K\left(\frac{x - x^{(i)}}{h}\right), \; x \in \mathbb{R},$$

where $(x^{(i)})_{i=1,\ldots,n}$ are the observed values of variable X_j and

$$K(x) := \int_{-\infty}^{x} \varphi(t) dt$$

is the Gaussian kernel. The bandwidth parameter $h > 0$ is chosen such that the asymptotic mean integrated squared error is minimized (see Duong (2016) for details). The pseudo-copula data

$$\hat{\mathbf{v}} = \left(\hat{F}_Y(y^{(i)})\right)_{i=1,\ldots,n} = \left(\hat{v}^{(i)}\right)_{i=1,\ldots,n}$$

$$\hat{\boldsymbol{\mathcal{U}}} = \left(\hat{F}_j(x_j^{(i)})\right)_{j=1,\ldots,d, i=1,\ldots,n} = \left(\hat{u}_j^{(i)}\right)_{j=1,\ldots,d, i=1,\ldots,n}$$

is then an approximately i.i.d. sample from the PIT random vector $(V, U_1, \ldots, U_d)^T$. In the second step, a D-vine copula is fitted to the pseudo-copula data.

3.5.2 Fitting a D-vine Copula to the Data

The ordering of the variables in the D-vine significantly controls the explanatory power of the resulting model. As the number of predictor variables increases, comparing all $d!$ possible orderings is soon infeasible. Therefore, Kraus and Czado (2017) present a greedy approach to identify the order of the D-vine. In the order, V is defined to be the first node, whereas the sequence of the predictor variables $U_1, \ldots, U_d$ is allowed to be arbitrary and is assigned on the fly by the algorithm. It sequentially selects the most influential predictor from all predictors yet not appearing in the D-vine to be next in the ordering. The most influential predictor is considered to be the predictor which would most increase the conditional log-likelihood of the resulting D-vine. If none of the remaining predictors is able to increase the conditional log-likelihood of the model, the algorithm stops and returns the current D-vine. This automatic forward covariate selection results in parsimonious models (Kraus and Czado, 2017). The conditional log-likelihood of an estimated D-vine copula is given as follows:

Definition 3.13 (Conditional log-likelihood of an estimated D-vine copula (Kraus and Czado, 2017))**.** Let us assume, a D-vine with order $V - U_{l_1} - \cdots - U_{l_d}$ has been estimated, where $\boldsymbol{l} = (l_1, \ldots, l_d)^T$ is the corresponding permutation of $\{1, \ldots, d\}$. Then, the conditional log-likelihood (cll) of the D-vine is defined as

$$\mathrm{cll}\big(\boldsymbol{l}; \hat{\mathrm{v}}, \hat{\boldsymbol{\mathcal{U}}}\big) := \sum_{i=1}^{n} \ln c_{V|\boldsymbol{U}}\big(\hat{v}^{(i)} \mid \hat{\boldsymbol{u}}^{(i)}; \boldsymbol{l}\big).$$

The conditional copula density $c_{V|\boldsymbol{U}}$ can be expressed as the product over all pair-copulas of the D-vine that contain V:

$$c_{V|\boldsymbol{U}}\big(\hat{v}^{(i)} \mid \hat{\boldsymbol{u}}^{(i)}; \boldsymbol{l}\big) = c_{VU_{l_1}}\big(\hat{v}^{(i)}, \hat{u}_{l_1}^{(i)}\big) \times$$

$$\prod_{j=2}^{d} c_{VU_{l_j};U_{l_1},\ldots,U_{l_{j-1}}}\big(\hat{C}_{V|U_{l_1},\ldots,U_{l_{j-1}}}(\hat{v}^{(i)}|\hat{u}_{l_1}^{(i)}, \ldots, \hat{u}_{l_{j-1}}^{(i)}), \hat{C}_{U_{l_j}|U_{l_1},\ldots,U_{l_{j-1}}}(\hat{u}_{l_j}^{(i)}|\hat{u}_{l_1}^{(i)}, \ldots, \hat{u}_{l_{j-1}}^{(i)}\big).$$

A proof of this can be found in Tepegjozova (2019). However, the conditional log-likelihood does not incorporate the model complexity. Therefore, Kraus and Czado (2017) suggest the AIC- and BIC-corrected log-likelihood to use as a selection criterion to achieve even more parsimonious models. Since the resulting D-vine may also contain nonparametric pair copulas, we do not have a number of parameters readily available. As a substitute, the degrees of freedom for a density estimation fit, as proposed in Loader (1999), is used as an *effective number of parameters*. Let $|\hat{\boldsymbol{\theta}}|$ be the number of parameters (including the effective number of parameters) used for the construction of the D-vine, the AIC- and BIC-corrected log-likelihood is defined as

$$\begin{aligned} \mathrm{cll}^{\mathrm{AIC}}\big(\boldsymbol{l}; \hat{\mathrm{v}}, \hat{\boldsymbol{\mathcal{U}}}\big) &:= -2\mathrm{cll}\big(\boldsymbol{l}; \hat{\mathrm{v}}, \hat{\boldsymbol{\mathcal{U}}}\big) + 2|\hat{\boldsymbol{\theta}}| \quad \text{and} \\ \mathrm{cll}^{\mathrm{BIC}}\big(\boldsymbol{l}; \hat{\mathrm{v}}, \hat{\boldsymbol{\mathcal{U}}}\big) &:= -2\mathrm{cll}\big(\boldsymbol{l}; \hat{\mathrm{v}}, \hat{\boldsymbol{\mathcal{U}}}\big) + \log(n)|\hat{\boldsymbol{\theta}}|. \end{aligned} \tag{17}$$

For a demonstration of the DVQR algorithm, we point out the k-th step in the model fit. Let us assume, the method has so far selected $k-1$ predictors and the current D-vine has the ordering $V - U_{l_1} - \cdots - U_{l_{k-1}}$, where $(l_1, \ldots, l_{k-1}) \subset (1, \ldots, d)$. For all predictors $U_j, j \in (1, \ldots, d) \backslash (l_1, \ldots, l_{k-1})$ not yet in the D-vine, the necessary pair-copulas to extend the current D-vine to $V - U_{l_1} - \cdots - U_{l_{k-1}} - U_j$ are estimated. Figure 3 illustrates this procedure. In each step of the DVQR method, AIC-optimal pair-copulas are chosen. More precisely, in case of parametric copulas, we minimize the AIC over all possible copula families and their corresponding parameter spaces to estimate the copula. Additionally, the AIC of an estimated nonparametric pair-copula is compared, where again, the effective number of parameters is applied in the definition of the AIC. The DVQR method also integrates an independence test at level 0.05 to check whether the two variables are independent of each other. In this case, the *independence copula* $C(u_i, u_j) = u_i \cdot u_j$ with constant copula density $c(u_i, u_j) = 1$ is assigned. It can be considered as a special case of the Gaussian copula where the correlation matrix equals the identity matrix I_2.

Having estimated the necessary pair-copulas to extend the current D-vine for each of the U_j, we can now compute the conditional log-likelihood of the model containing U_j. If none

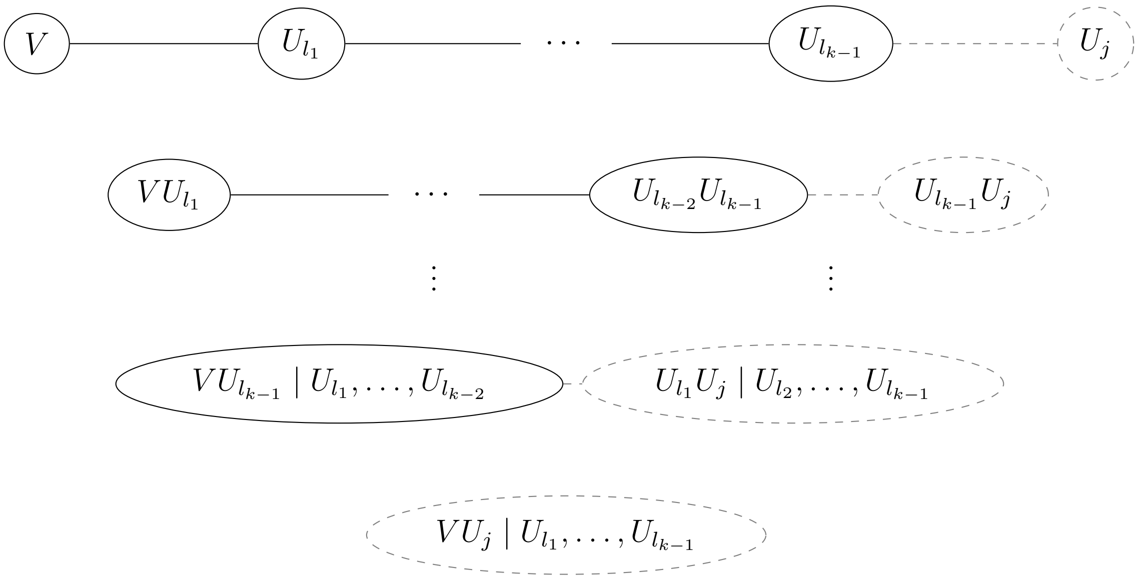

V
U_{l_1}
$\cdots$
$U_{l_{k-1}}$
U_j
VU_{l_1}
$\cdots$
$U_{l_{k-2}}U_{l_{k-1}}$
$U_{l_{k-1}}U_j$
$VU_{l_{k-1}} \mid U_{l_1}, \ldots, U_{l_{k-2}}$
$U_{l_1}U_j \mid U_{l_2}, \ldots, U_{l_{k-1}}$
$VU_j \mid U_{l_1}, \ldots, U_{l_{k-1}}$

Figure 3: Illustration of the k-th step in the DVQR algorithm, where the current D-vine is extended with another predictor variable

of the resulting models achieves an increase of the conditional likelihood, the algorithm stops. Otherwise it updates the model by adding the variable corresponding to the highest conditional log-likelihood (or the lowest AIC- or BIC-corrected version thereof).

Through the construction of the D-vine quantile regression the typical shortfalls that linear quantile regression involves are obviated. For example, the conditional copula quantile function in Equation (16) is monotonically increasing in α. Therefore, a crossing of quantiles functions for different α levels cannot occur (Kraus and Czado, 2017). Furthermore, DVQR poses less restrictive model assumptions since it allows for a more flexible modelling of the dependence structure. It does not assume linearity of the conditional quantiles as in linear quantile regression. If we, for instance, assume normal margins, the Gaussian copula is the only copula that would result in linear conditional quantiles.

3.6 Performance Measures

In order to assess the quality of predicted quantiles and prediction intervals we employ two different performance measures. The interval score by Gneiting and Raftery (2007) is a suitable scoring rule for $(1-\alpha) \times 100\%$ prediction intervals in terms of narrowness and accuracy of the intervals. It rewards narrow intervals and, at the same time adds a penalty, the size of which depends on α, if an observation does not fall into the interval. The smaller the interval score the better.

Definition 3.14 (Interval score (Gneiting and Raftery, 2007))**.** Let $\hat{u}_i^{\mathcal{M}}$ and $\hat{\ell}_i^{\mathcal{M}}$ be the upper and lower predictive quantiles at level $\alpha/2$ by model $\mathcal{M}$ for the i-th observation. For the central $(1-\alpha)\times 100\%$ prediction interval, the interval score of model $\mathcal{M}$ is defined as

$$\mathrm{IS}(\mathcal{M}) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left[(\hat{u}_i^{\mathcal{M}} - \hat{\ell}_i^{\mathcal{M}}) + \frac{2}{\alpha}(\hat{\ell}_i^{\mathcal{M}} - y_i)\mathbb{1}_{y_i < \hat{\ell}_i^{\mathcal{M}}} + \frac{2}{\alpha}(y_i - \hat{u}_i^{\mathcal{M}})\mathbb{1}_{y_i > \hat{u}_i^{\mathcal{M}}} \right].$$

The interval score considers a central prediction interval and therefore, incorporates two predicted quantiles. In case of linear quantile regression, quantile crossing poses a problem that the above definition does not address. Accordingly, we customize the definition of the interval score by adding a penalty if quantile crossing occurs:

$$\begin{aligned} \mathrm{IS}(\mathcal{M}) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \Bigg[&(\hat{u}_i^{\mathcal{M}} - \hat{\ell}_i^{\mathcal{M}})\mathbb{1}_{\hat{u}_i^{\mathcal{M}} > \hat{\ell}_i^{\mathcal{M}}} + \frac{2}{\alpha}(\hat{\ell}_i^{\mathcal{M}} - \hat{u}_i^{\mathcal{M}})\mathbb{1}_{\hat{\ell}_i^{\mathcal{M}} > \hat{u}_i^{\mathcal{M}}} \\ &+ \frac{2}{\alpha}(\hat{\ell}_i^{\mathcal{M}} - y_i)\mathbb{1}_{y_i < \hat{\ell}_i^{\mathcal{M}}} + \frac{2}{\alpha}(y_i - \hat{u}_i^{\mathcal{M}})\mathbb{1}_{y_i > \hat{u}_i^{\mathcal{M}}} \Bigg]. \end{aligned} \tag{18}$$

The second performance measure is based on the check loss of a predicted quantile. The check loss function has also been applied as a loss function in the derivation of the linear quantile regression method (see Section 2.2). It asymmetrically weighs observations that lie above and below the predicted quantile. Observations that lie on the "smaller" side of the quantile receive larger weights and vice versa.

Definition 3.15 (Check loss)**.** Let $\hat{q}_{\alpha,i}^{\mathcal{M}}$ be the predictive quantile at level α by model $\mathcal{M}$ for the i-th observation. The check loss of a model $\mathcal{M}$ and a α-quantile is then given as

$$\begin{aligned} \mathrm{CL}_\alpha(\mathcal{M}) :=& \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \rho_\alpha(y_i - \hat{q}_{\alpha,i}^{\mathcal{M}}) \\ =& \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (y_i - \hat{q}_{\alpha,i}^{\mathcal{M}})(\alpha - \mathbb{1}_{y_i - \hat{q}_{\alpha,i}^{\mathcal{M}} < 0}). \end{aligned}$$

Again, smaller values are better.

Having revisited the theory behind the D-vine copula-based and linear quantile regression method, we will now continue with the application of these two methods to the Seoul weather data. At first, we will describe the data. Then, we will explicate the models and the training process. Finally, the results will be compared and evaluated.

4 Application

We now attend to the Seoul weather data and the estimation of prediction intervals for next-day air temperatures. The data in our analysis originates from the UCI machine learning repository (*https://archive.ics.uci.edu/ml/index.php*) and was first studied by Cho et al. (2020). They shared their data for free download. It contains weather data of 25 different weather stations in the urban area of Seoul. For a location of the weather stations see Figure 4. From 2013 to 2017, data has been collected between June 30 and August 30, which makes a total of 62 days of recorded weather data per year. In total the dataset consists of $5 \times 25 \times 62 = 7,750$ observations, remaining $7,588$ when excluding missing values. Originally, Cho et al. (2020) aim at enhancing next-day maximum and minimum air temperature forecasts of a local weather model over Seoul, the so-called LDAPS model[*]. The LDAPS model outputs next-day predictions of several weather quantities, such as maximum and minimum air temperature, average wind speed and cloud cover. Cho et al. (2020) use these predictions as input variables in their post-processing models. Additionally, they include the present-day maximum and minimum air temperature as predictors as well as a few geographical and topological variables which describe the location of the different weather stations. In the following section, we will give a short insight into numerical weather prediction, its challenges and the common practice to post-process NWP models.

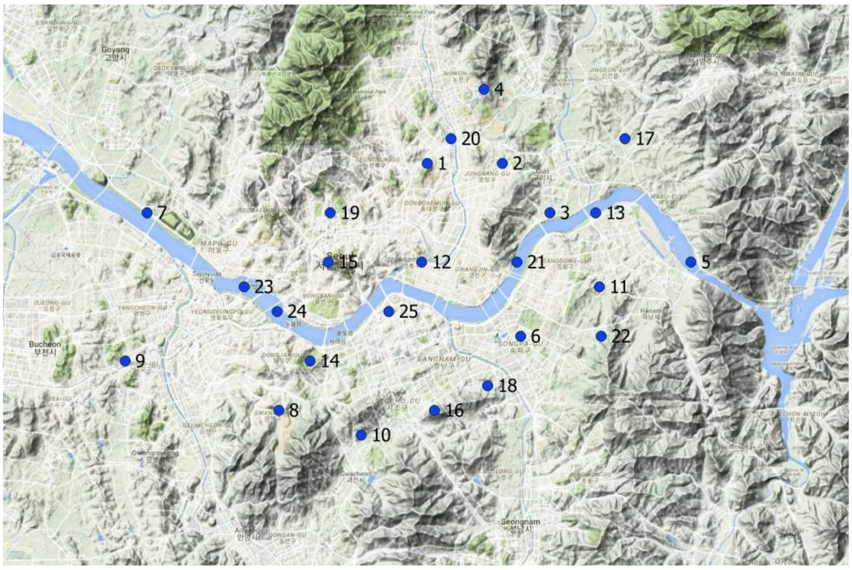


Figure 4: Location of weather stations

4.1 Numerical Weather Prediction Models

In times of climate change and considerable accumulation of extreme weather events all across the globe, the need for accurate weather predictions is self-evident. Extreme

[*] The **LDAPS** model (Local Data Assimilation and Prediction System) is a local **NWP** (Numerical Weather Prediction) model over South Korea.

weather events can cause economic, social and environmental problems such as health issues, bush fires or flooding. In particular, precise air temperature forecasts are essential in order to take appropriate precautions against the health-related risks of heat waves and cold spells. In the early years of the 20th century, well before the first computer existed, Lewis F. Richardson was the first to formulate the problem of weather prediction numerically (Lynch, 2006). He provided a fundamental basis for all further research in numerical weather prediction models.
Numerical weather prediction attempts to predict measurable weather quantities through solving complex sets of partial differential equations that describe the dynamics of the atmosphere and the oceans (Bacmeister, 2012). It was not until the emergence of powerful computers that NWP models could produce reasonable predictions. However, these models are highly sensitive to the initial conditions of the atmosphere which are modelled based on observed quantities. The uncertainty of physical parametrizations in the NWP model further limits its prediction power (Cho et al., 2020; Bacmeister, 2012). Therefore, various post-processing methods are utilized to increase forecasting accuracy.

Cho et al. (2020) have developed and evaluated four different machine-learning-based methods to post-process the output of the LDAPS model. The LDAPS model was designed by the Korea Meteorological Administration (KMA) and is based on the United Model of the UK Met Office (Cho et al., 2020). Initially, it was constructed to overcome the limitations of two other NWP models that the KMA has been operating, the Global and Regional Data Assimilation and Prediction models (GDAPS and RDAPS). Due to their coarse grid resolution these models fail to capture extreme weather events. Still, the LDAPS model output typically suffers a systemic bias. With their post-processing methods, Cho et al. (2020) could successfully reduce the bias of the LDAPS model's next-day maximum and minimum air temperature forecast. In the following, we will draw our attention to estimate prediction intervals rather than exact temperature values. We utilize the data to comparatively investigate prediction ability of D-vine copula-based and linear quantile regression on a sizeable dataset.

4.2 Inspection and Pre-Processing of the Seoul Weather Data

We will now take a closer look at the data. It consists of summer data of five years, ranging from 2013 to 2017, with 21 predictor variables and next-day maximum and minimum air temperature as response variables. Cho et al. (2020) divide the predictors into three groups. There are

- 14 predictor variables originating from the LDAPS model output,
- 5 variables that describe topological and geographical properties of the weather stations and
- 2 in-situ variables, i.e. maximum and minimum air temperature between 0-21 h of the present day.

More precisely, the first set of predictor variables consist of the LDAPS model forecasts of several next-day weather measures such as *maximum* and *minimum air temperature*, *maximum* and *minimum relative humidity*, *average wind speed* and *latent heat flux*. In addition, *average cloud cover* and *average precipitation* are forecasted. For that matter, each day is split into four time sections — (0-5 h), (6-11 h), (12-17 h), (18-23 h) — where average cloud cover and average precipitation is predicted for each of these 6-hour splits separately. All of these variables take on a wide range of values and are therefore considered continuous.

As a second set of predictor variables, there are station-specific factors. *Latitude* and *longitude* describe the exact geographical coordinates of the weather station. The variables *Slope* and *DEM* indicate the slope and elevation of the respective weather station. Lastly, the daily incoming *solar radiation* is measured and stored for each weather station. Since the first four variables only depend on the respective weather station and not on time, each of these variables take on a maximum of 25 different values. This does not apply to solar radiation as it depends on the specific day. However, we also consider all of these variables to be continuous. This will potentially affect performance of the D-vine regression which we will address later. As we have mentioned in the beginning of Chapter 3, Schallhorn et al. (2017) have adapted the D-vine quantile regression approach of Kraus and Czado (2017) to also allow for discrete or factor variables as input. This is also implemented in the `vinereg()` method. However, it would not be reasonable to use the station-specific variables as factors since then, the scale of the variables would be lost.
Furthermore, *present-day maximum* and *minimum air temperature* also serve as predictor variables. These quantities are only measured between 0-21 h of the present day in order to give time to the necessary computation of next-day predictions.

In the following, we will provide a more detailed description of the predictor variables and introduce new predictor variables which we create by aggregating or modifying some of the existing ones. At first, let us inspect the response variables. Naturally, we have $\textit{next-day } T_{min} \leq \textit{next-day } T_{max}$. This would require restrictions of the resulting copula. This particularly poses a problem if the two response variables are modelled simultaneously (cf. notes on further research in Section 5). As a remedy, we introduce a new response variable by taking the difference between the next-day maximum and minimum air temperature instead of choosing next-day maximum air temperature as a response:

$$\textit{next-day } T_{diff} := \textit{next-day } T_{max} - \textit{next-day } T_{min}.$$

Now, $\textit{next-day } T_{min}$ and $\textit{next-day } T_{diff}$ do not have any obvious relationship. Estimating prediction intervals for *next-day minimum* and *difference air temperature* is now the objective of our application. To get a first glimpse of the temperature profile, we consider the temperature values over time. We choose weather station 25 for an exemplary temperature profile, which is located centrally in the urban area of Seoul.
From the plot we see that the developing of temperature values does not reveal any exact patterns that occur in each year's temperature profile alike. Though, for the *next-day minimum temperature* the rough course of values shows slight analogies for 2013 and 2016 as well as for the remaining years. Nevertheless, we will not specifically take account of seasonal effects in the regression analysis, but indirectly through a training set selection (cf. Section 4.4.1).

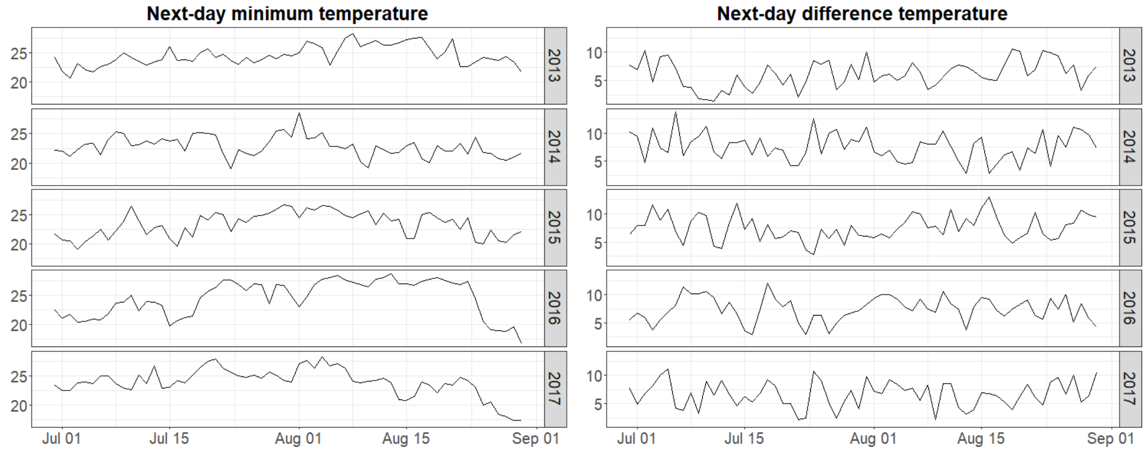


Figure 5: Yearly temperature profile of weather station 25

For the in-situ variables *present-day maximum* and *minimum air temperature* and the LDAPS model forecasts *maximum* and *minimum air temperature* and *maximum* and *minimum relative humidity*, we likewise generate difference values:

$$\textit{Present-day } T_{\textit{diff}} := \textit{Present-day } T_{\textit{max}} - \textit{Present-day } T_{\textit{min}}$$

$$\textit{LDAPS } T_{\textit{diff}} := \textit{LDAPS } T_{\textit{max}} - \textit{LDAPS } T_{\textit{min}}$$

$$\textit{LDAPS } RH_{\textit{diff}} := \textit{LDAPS } RH_{\textit{max}} - \textit{LDAPS } RH_{\textit{min}}$$

Regarding units of these variables, we have degree Celsius (°C) for all variables which represent temperature values. Relative humidity values range from 0 to 100 representing percentage. On the Earth's surface, insolation leads to the evaporation of water. Since it does not result in a temperature rise, the expended energy is stored in the gaseous water. Once the ascending water condensates in the atmosphere, the energy is released in terms of heat. The *average latent heat flux* describes the average flux of this latent energy and is measured in Watt per square metre (W/m^2). Similarly, *solar radiation* describes the energy per unit area that reaches the Earth's surface in form of electromagnetic radiation from the sun. Since it is measured over the course of the day, it is expressed in watt-hour per square metre (wh/m^2). The *average wind speed* is measured in metre per second (m/s). All of these variables enter our regression analysis as they are. Next, we aggregate the LDAPS 6-hour split variables regarding cloud cover and precipitation in order to reduce model complexity. The *average cloud cover* is given as a ratio where 0 implies a clear sky with no clouds at all and 1 a completely overcast sky. At first, we transform the cloud cover variables using the logit transform:

$$\operatorname{logit}(c) := \ln\left(\frac{c}{1-c}\right), \quad \text{for } c \in (0,1).$$

To avoid any issues at the boundaries, the values of the cloud cover variables are mapped to $[0.005, 0.995]$ prior to transforming. Through the logit transformation the cloud cover variables are now supported on full $\mathbb{R}$, which is advantageous for the kernel density estimation of the marginals. Secondly, we merge the four 6-hour split variables into a day-time (6-17 h) and a night-time (18-5 h) cloud cover variable:

$$\begin{aligned} CC_{night} &:= \Big(\text{logit}\big(CC_{\textit{0-5 h}}\big) - \text{logit}\big(CC_{\textit{18-23 h}}\big)\Big)\Big/2, \\ CC_{day} &:= \Big(\text{logit}\big(CC_{\textit{6-11 h}}\big) - \text{logit}\big(CC_{\textit{12-17 h}}\big)\Big)\Big/2. \end{aligned}$$

The *night-time cloud cover* and *next-day difference temperature* appear to have a fairly strong negative correlation. This seems reasonable since temperatures usually drop comparatively lower in starlit nights than during overcast nights. The precipitation variables are measured in litre per square metre (l/m^2). The measure of precipitation involves the issue of quite a few zero entries in the data. This poses a problem for the estimation of the marginals and we would not obtain uniformly distributed marginals for the pseudo-copula data. On 36% of the recorded days, there was no precipitation at all. Additionally, on many days, rainfall was only observed for one or two 6-hour splits. Instead of using the original precipitation variables we create a variable PPT_{avg} that takes the average over the four 6-hour splits and PPT_{max} that takes the maximum 6-hour split. To account for the duration of the rainfall, we also create two new variables that consider the number of rainy 6-hour splits. As a binary variable, *seq.rain*$_{\geq 3}$ indicates if there were at least 3 rainy 6-hour splits in a row:

$$\textit{seq.rain}_{\geq 3} := \begin{cases} 1 & , PPT_{\textit{0-5 h}}, PPT_{\textit{6-11 h}}, PPT_{\textit{12-17 h}} > 0 \\ 1 & , PPT_{\textit{6-11 h}}, PPT_{\textit{12-17 h}}, PPT_{\textit{18-23 h}} > 0 \\ 1 & , PPT_{\textit{0-5 h}}, PPT_{\textit{6-11 h}}, PPT_{\textit{12-17 h}}, PPT_{\textit{18-23 h}} > 0 \\ 0 & , \text{else} \end{cases}$$

As we have mentioned before, the `vinereg()` method can also handle binary predictor variables. However, in the following, we will see that *seq.rain*$_{\geq 3}$ does not play a role in the DVQR models. Therefore, we did not explicitly address the case of binary input variables in the theoretical part of this thesis. Further explanation of how discrete variables are handled in DVQR can be found in Schallhorn et al. (2017). As a last variable regarding precipitation, we introduce *rain* × *sections* which multiplies the daily sum of precipitation by the number of rainy 6-hour splits:

$$\textit{rain} \times \textit{sections} := \Bigg[\sum_{\substack{i\in\{\textit{0-5 h},\textit{6-11 h},\\ \textit{12-17 h},\textit{18-23 h}\}}} PPT_i\Bigg] \times \Bigg[\sum_{\substack{i\in\{\textit{0-5 h},\textit{6-11 h},\\ \textit{12-17 h},\textit{18-23 h}\}}} \mathbb{1}_{PPT_i>0}\Bigg].$$

Thus, this variable gives more weight to continuous rain which might have a greater effect on temperature than a short rain shower. On the left panel of Figure 6, the daily precipitation at weather station 25 is depicted. The right panel shows the distribution of the number of rainy 6-hour splits, partitioned by years.

For a first visual exploration of the dependence structure among the response variables and the predictors, we have produced empirical normalized pair contour plots of the Seoul

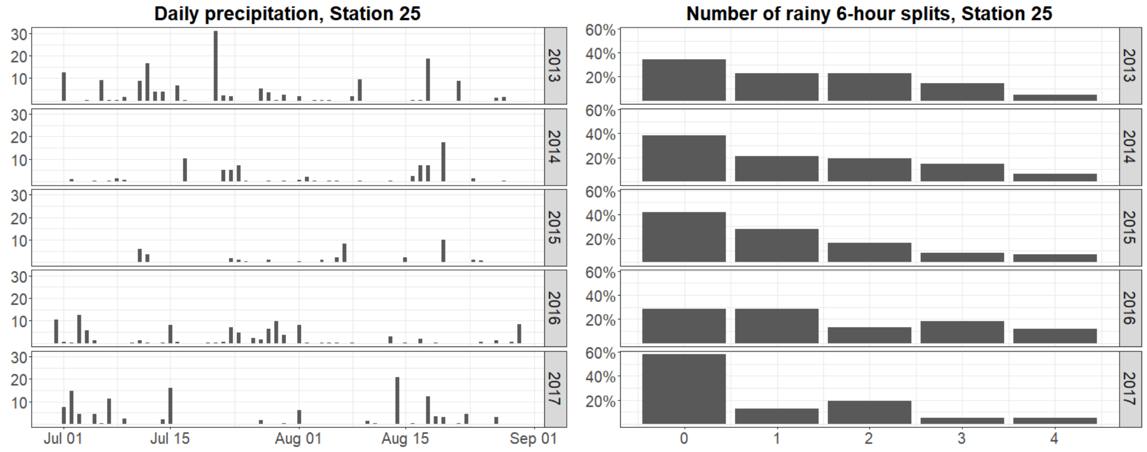
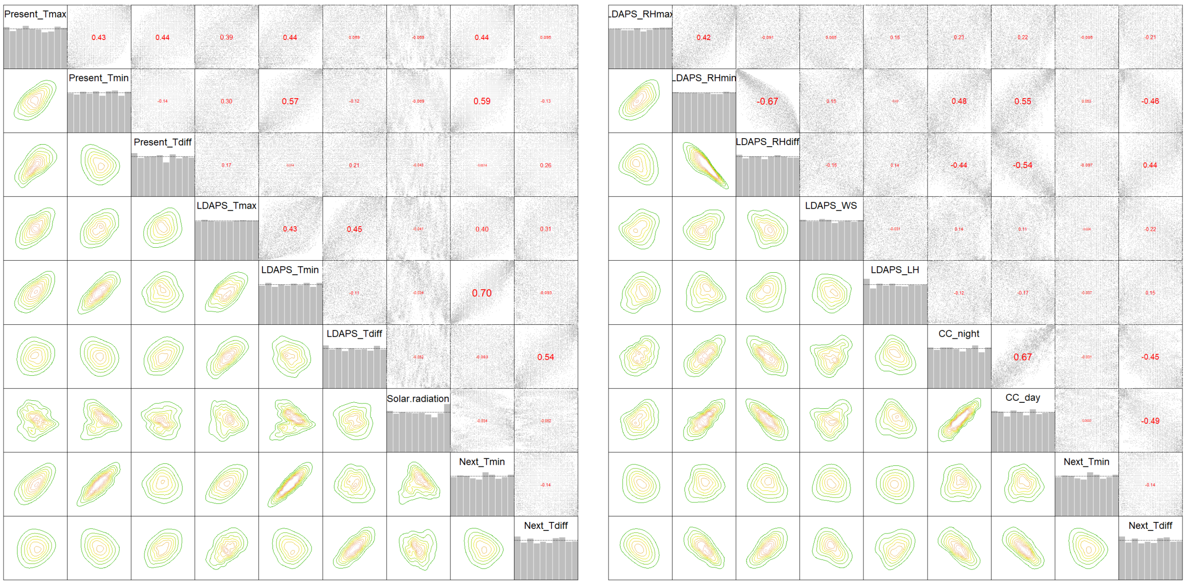


Figure 6: Daily precipitation and number of rainy 6-hour splits at station 25

weather data (see Figure 7) by applying the generic `pairs()` function to a `copuladata` object (see Section 3.1.3 for a detailed introduction of normalized bivariate contour plots). Such an object contains our input data transformed into pseudo-copula data. Therefore, we first have to estimate marginal densities using the continuous kernel smoothing estimator from Section 3.5.1. The `pairs()` function then generates a matrix of plots where on the upper triangle we have pairs plots of the pseudo-copula data. On the diagonal, marginal histograms of the pseudo-copula data are displayed. On the lower triangle, we have empirical contour plots of the normalized pseudo-copula data.

For better presentability, we have split the pair contour plots into two parts. Further, we have excluded the four station-dependent location variables that describe elevation, slope, longitude and latitude of the stations. Since they only take 25 different values each, the normalized pair contour plots would not produce any interpretable results. This also applies to the precipitation variables due to their large number of zero values. We can already observe some interesting contour shapes and deviations from the Gaussian copula including asymmetries and tail dependencies. For example, the contours of *present-day* T_{min} and *present-day* T_{diff} indicate a Clayton copula. The dependence structure between *next-day* T_{diff} and CC_{night} or CC_{day} could be represented by a Frank copula, while for *next-day* T_{min} and CC_{night} or CC_{day} it rather looks like a nonparametric copula shape. Especially the pair contours of *solar radiation* with other variables also reveal nonparametric copula shapes. Some pairs of variables seem to exhibit tail dependence, for instance *next-day* T_{min} and *present-day* T_{min}. As expected, we observe strong dependence between *next-day* T_{min} and *LDAPS* T_{min} showing approximately a Gaussian copula shape. Similarly, *next-day* T_{diff} and *LDAPS* T_{diff} have a strong relationship that also exhibits approximately Gaussian copula shape.



(a)

(b)

Figure 7: Empirical normalized pair contour plots of Seoul weather data including all summer periods from 2013 to 2017

4.3 Model Building and Selection

Performing DVQR on large data with numerous predictor variables is fairly time-consuming due to the large number of pair copulas that have to be estimated. In the context of temperature predictions, we will separately train a model for each day that we want to predict. Therefore, we aim at finding parsimonious models and in particular, a threshold for the number of predictors such that the model's explanatory power does not significantly change beyond that number. Prior to training models for next-day predictions, we perform a model selection where we try to identify the most influential predictor variables. Our model selection will be based on the first four years of the given data, thus, excluding 2017 which will be used for validating our predictions (see Section 4.4 for estimation of prediction intervals). For the two response variables, we fit distinct models. Since we have introduced new predictor variables which represent differences of other variables, these variables are no longer linearly independent. In each case, we will choose two out of three linearly dependent predictors according to their highest pairwise empirical Kendall's τ values with the response. These values are given in Figure 8. Furthermore, we reduce the number of cloud cover variables by replacing the four 6-hour split variables by the aggregated day-time and night-time variables. We also replace the four precipitation 6-hour split variables with our newly generated variables. Lastly, we include for each response the most influential 6-hour split precipitation variable according to their empirical Kendall's τ values (cf. Figure 8). For each response variable, we end up with a set of 20 eligible predictor variables that enter the model. Table 4 in Appendix A.1 gives a list of these variables. For the `vinereg()` method, we set the selection criterion to *"bic"* (see Equation (17)). Thus, variables are selected based on the BIC-corrected conditional log-likelihood resulting in more parsimonious models compared to the *"aic"* and *"cll"* selection criterion. Since we like to allow for nonparametric copulas in the D-vine density, we set the *family_set* argument to *"all"*. As described in Section 3.5, the algorithm will automatically discard predictors according to the BIC selection criterion. All remaining predictors appear in the order of the fitted D-vine and we readily obtain an ordering of the predictors according to their explanatory power. One at a time, we will then remove the last predictor in the order of the D-vine and again fit a new model. Thereby, we obtain a set of nested models. At the same time, we fit a set of nested LQR models. Analogue to the automated forward selection of variables in the D-vine regression, we establish a BIC-based forward selection for LQR. Starting with zero predictors in the model, in each step, we add the predictor variable that would most lower the BIC of the resulting model. If none of the remaining predictors results in a lower BIC, we stop. We use the same pool of eligible predictors as for DVQR. A detailed list of these variables and their corresponding ordering in the DVQR and LQR can respectively be found in Table 4 from Appendix A.1.

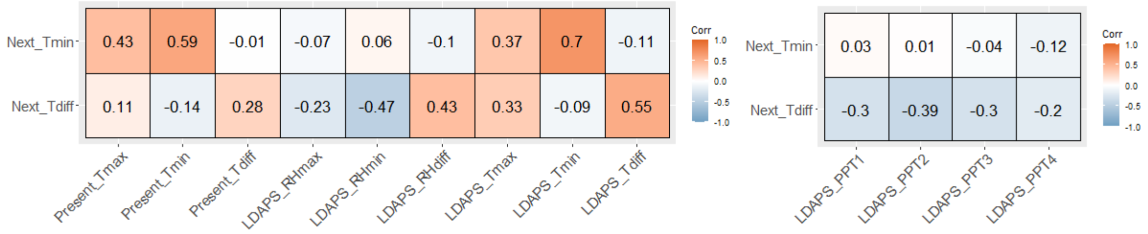


Figure 8: Empirical pairwise Kendall's τ values

In the upper panel of Figure 9, the BIC for the sets of nested models is displayed. The horizontal axis shows the number of included predictors. For *next-day* T_{min}, the LQR already stops at a number of 11 predictors. The BIC cannot be reduced by any of the remaining predictors. The DVQR, in contrast, takes almost all predictors into the model. For *next-day* T_{diff}, both DVQR and LQR already eliminate a few predictors from the model. LQR stops at 13 predictors, DVQR at 14. At first, we will use a threshold of 10 for the model size in our prediction models. The model fit does not significantly increase beyond that number and we can rigorously reduce computation time. At the same time, reducing the number of predictors lowers the risk of overfitting. Later in our application, we will also examine a more restrictive threshold of five for the number of predictor variables in the model. In the lower panel of Figure 9, the in-sample interval score for the central $99\%, 95\%$ and 90% prediction interval is displayed. For the 95% and 90% prediction intervals, we observe that the DVQR achieves slightly lower interval scores compared to LQR.

Finally, we will look at the first 10 predictors in the LQR and DVQR order, respectively. We will use these predictors in our prediction models in the next section. Table 2 lists these variables in the corresponding orders of the fitted D-vine and linear quantile regression model for *next-day* T_{min} and *next-day* T_{diff}, respectively. As expected, we observe that in all cases the LDAPS model's prediction of the next-day temperature value seem to be the

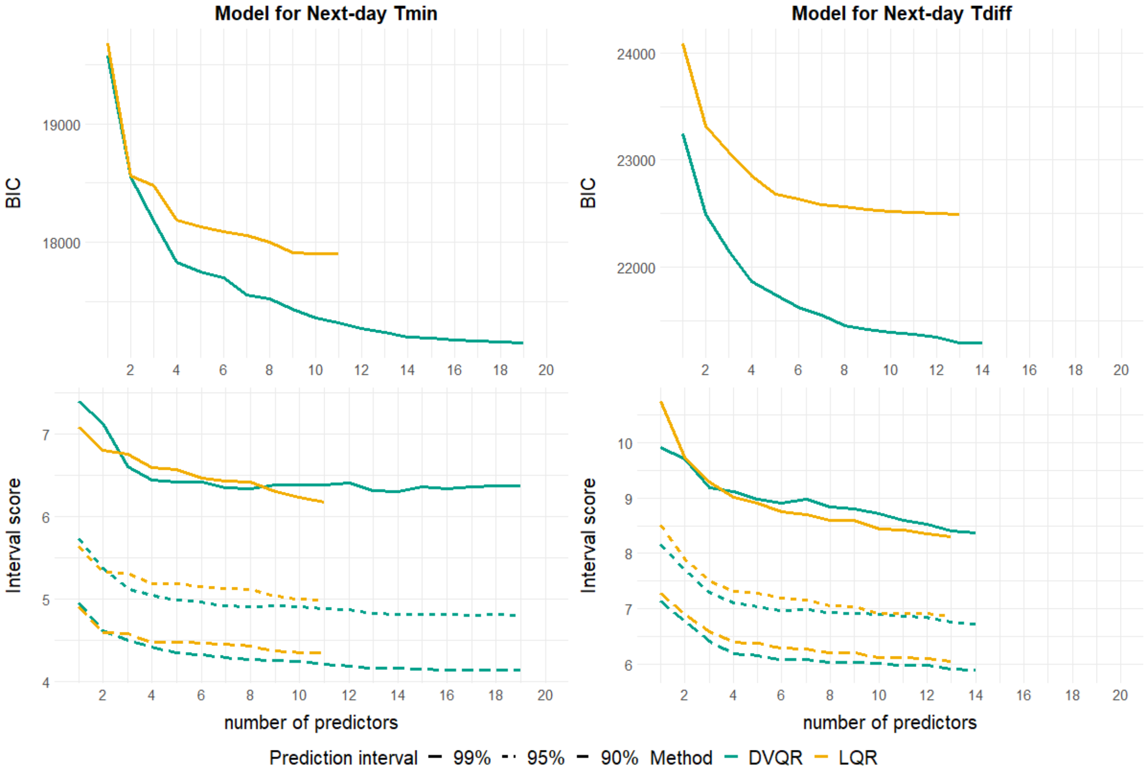


Figure 9: BIC and in-sample interval score for three different prediction intervals with respect to the number of predictors included in the model

most influential predictor. Interestingly, in the DVQR, the dependence between *next-day* T_{min} and *LDAPS* T_{min} is modelled by a Gaussian copula, while the dependence between *next-day* T_{diff} and *LDAPS* T_{diff} is estimated through a nonparametric copula. Since the LQR also assumes the dependence structure between the response and the predictors to be Gaussian, we do not expect DVQR to outperform LQR in this one predictor case. For a DVQR model with 10 predictors, we have $\sum_{i=1}^{10} i = 55$ pair copulas in the D-vine density. For *next-day* T_{min}, roughly one third of these pair copulas were estimated to be nonparametric and Archimedean copulas, respectively. For *next-day* T_{diff}, also one third of the pair copulas were estimated nonparametrically, while almost every second pair copula was modelled as an Archimedean copula. This indicates that, in general, using only Gaussian or elliptical copulas would not provide reasonable fits.

While for *next-day* T_{min} only the first two predictors coincide for the DVQR and LQR model, the orders of predictors equal up to the seventh position for *next-day* T_{diff}. We also ascertain the large influence of CC_{night} on *next-day* T_{diff} as it appears at the second position for both DVQR and LQR.

order	DVQR	LQR
1	LDAPS $\mathrm{T_{min}}$	LDAPS $\mathrm{T_{min}}$
2	Present-day $\mathrm{T_{min}}$	Present-day $\mathrm{T_{min}}$
3	Present-day $\mathrm{T_{max}}$	Elevation
4	Longitude	Slope
5	LDAPS WS	rain × sections
6	Slope	LDAPS $\mathrm{RH_{max}}$
7	Elevation	$\mathrm{seq.rain_{\geq 3}}$
8	rain × sections	LDAPS $\mathrm{RH_{diff}}$
9	LDAPS $\mathrm{RH_{max}}$	LDAPS $\mathrm{T_{max}}$
10	$\mathrm{CC_{day}}$	LDAPS WS

(a) Response: Next-day $\mathrm{T_{min}}$

order	DVQR	LQR
1	LDAPS $\mathrm{T_{diff}}$	LDAPS $\mathrm{T_{diff}}$
2	$\mathrm{CC_{night}}$	$\mathrm{CC_{night}}$
3	Present-day $\mathrm{T_{diff}}$	Present-day $\mathrm{T_{diff}}$
4	LDAPS WS	LDAPS WS
5	LDAPS LH	LDAPS LH
6	Elevation	Elevation
7	Solar radiation	$\mathrm{CC_{day}}$
8	$\mathrm{CC_{day}}$	rain × sections
9	Longitude	Longitude
10	Slope	$\mathrm{PPT_{max}}$

(b) Response: Next-day $\mathrm{T_{diff}}$

Table 2: Selected predictors for a threshold of 10 predictors

4.4 Prediction

Based on data from 2013-2016, we have now identified for each response and regression model seemingly the 10 most influential predictor variables (cf. Table 2). With this, we will now investigate the predictive ability of the D-vine and linear quantile regression on 2017's data. Therefore, for each regression method and response, we predict conditional quantiles for next-day temperatures. For each day that we predict, we train a separate model.

4.4.1 Training Approaches

For next-day temperature quantile predictions, we examine three different training approaches. For comparability, we adapt the design of the training approaches such that we obtain equally sized training sets. In each case, the training set consists of 31 days, adding up to $31 \times 25 = 775$ observations. The first two approaches use a rolling training window which moves according to the day that we want to predict. The third approach randomly picks past days to include in the training set. Due to the limited availability of data and the design of the training approaches we can only comparably predict 28 successive days in 2017, ranging from July 31 to August 27.

1. Simple rolling window: This approach uses the past 31 successive days prior to the predicted day. Hence, the training set hardly changes from one day to another exchanging only one day at a time. Since we have data available only between June 30 and August 30 of each year, we initialize the rolling window training set with data from June 30, 2017

- July 30, 2017. Therefore, the first day that can be forecast like this is July 31.

2. Refined rolling window: In the context of temperature forecasting, this approach has been proposed by Möller et al. (2018). Instead of using a continuous rolling window as before, we now consider also several narrower windows around the predicted day from previous years. In particular, let us assume, we wish to forecast temperatures on day t in 2017. Then, for a natural number n, the refined rolling window will be composed of $\{t-n,\dots,t-2,t-1\}$ in 2017, and $\{t-n,\dots,t-2,t-1,t,t+1,t+2,\dots,t+n\}$ in each of the previous four years. The number n controls the size of the training set. We use a refined rolling window with $n=3$, such that the training set adds up to 31 days, the same size as for the simple rolling window. Due to this training approach, prediction is only feasible up to August 27.

3. Random drawing: In accordance to the training set size of the previous two approaches, we randomly draw 31 days from all past days prior to the predicted day to obtain a training set. That may also include days from previous years.

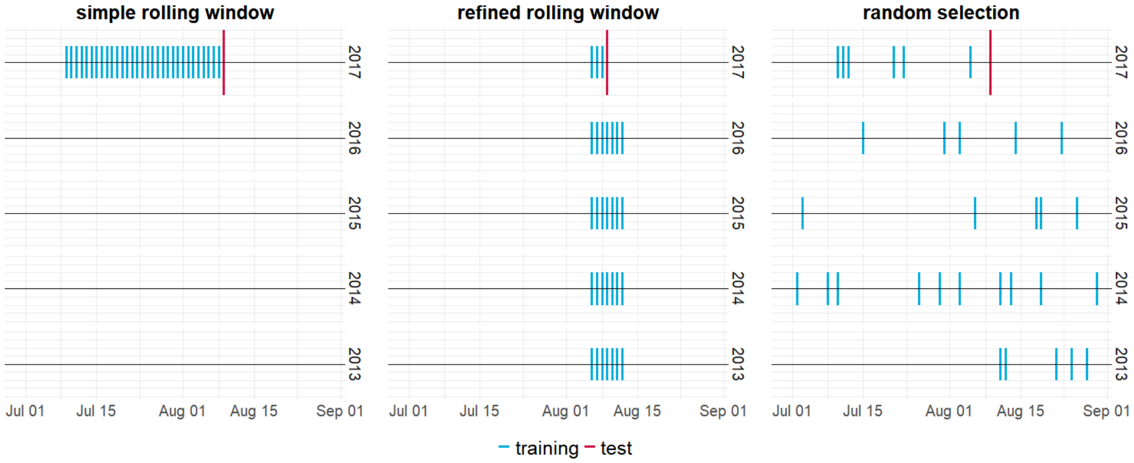


Figure 10: Visualization of the three different training approaches

Thus, for each training approach we train 28 separate models. The three different training approaches are visualized in Figure 10. Not only do we apply different training approaches but also consider two separate adjustments in the model fit.

4.4.2 Model Types

For each training approach, we now fit both D-vine and linear quantile regression models. In doing so, we allow for two different kind of model fits. As a first variant, we take each 10 predictors (from Table 2) into the model and do not allow for an automated variable selection. In particular, we preset the order of the D-vine as the same order that has been specified through the model selection (cf. Table 2). As a second variant, we allow for an automated BIC-based variable selection from the pre-chosen 10 predictors as described in

Section 3.5. In this case, it is also possible that the resulting model contains less than 10 predictors. In the following, we list the four different model notations:

- **full DVQR**: D-vine regression model that allows for both parametric and nonparametric copula estimation. The order of the D-vine is preset. It is the same order as the one that has been specified through the model selection process for the DVQR models.
- **dynamic DVQR**: Same as *full DVQR*, but now we do not preset the order of the D-vine. It is determined by the automated BIC-based forward selection for each training instance anew.
- **full LQR**: Linear quantile regression model including 10 preset predictor variables that have been specified through the model selection process for the LQR models.
- **dynamic LQR**: Analogue to *dynamic DVQR*. The linear quantile regression model allows for an automated BIC-based variable selection from the 10 pre-chosen predictors.

For each training approach and model type, we predict conditional quantiles at the following nine α-levels: $\{0.005, 0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975, 0.995\}$. With this, we can particularly estimate the $99\%, 95\%$ and 90% prediction interval. In total, we have 28 predicted days while in all instances the training set has a constant size of 31 days.

4.5 Results

4.5.1 Different Training Approaches with Fixed Training Horizon

At first, we juxtapose the interval scores of each model type and training approach for the $99\%, 95\%$ and 90% prediction interval in Figure 11. In addition to the interval score, we explicitly show the percentage of observations that fall into the predicted interval (cf. Figure 12). Since missed observations are accounted for in the definition of the interval score through a penalization, the results correlate. However, observations that lie far from the predicted interval as well as very broad intervals can also cause a relatively high interval score. In this case, the percentage of hits can still be high. Therefore, to have a more complete picture of the quality of the prediction intervals, we additionally look at the average width of the predicted intervals. In the computation of the width, we also have to take care of quantile crossing. A straightforward subtraction of the lower quantile from the upper quantile would yield a negative width for theses instances. Hence, we simply assign the maximal width of all predicted intervals to these instances which, in this way, imposes a form of penalty for quantile crossing. The average width of the intervals is shown in Figure 13. However, in our application, quantile crossing does not seem to be a great issue since it rarely occurs. For the simple rolling window, quantile crossing appeared in approx. 1.2% of all prediction instances and in approx. 0.4% for the random drawing. It did not at all occur for the refined rolling window.

First of all, we note that predictions for *next-day* T_{min} are generally more accurate than for *next-day* T_{diff}, thus resulting in lower interval score values. We will now separately

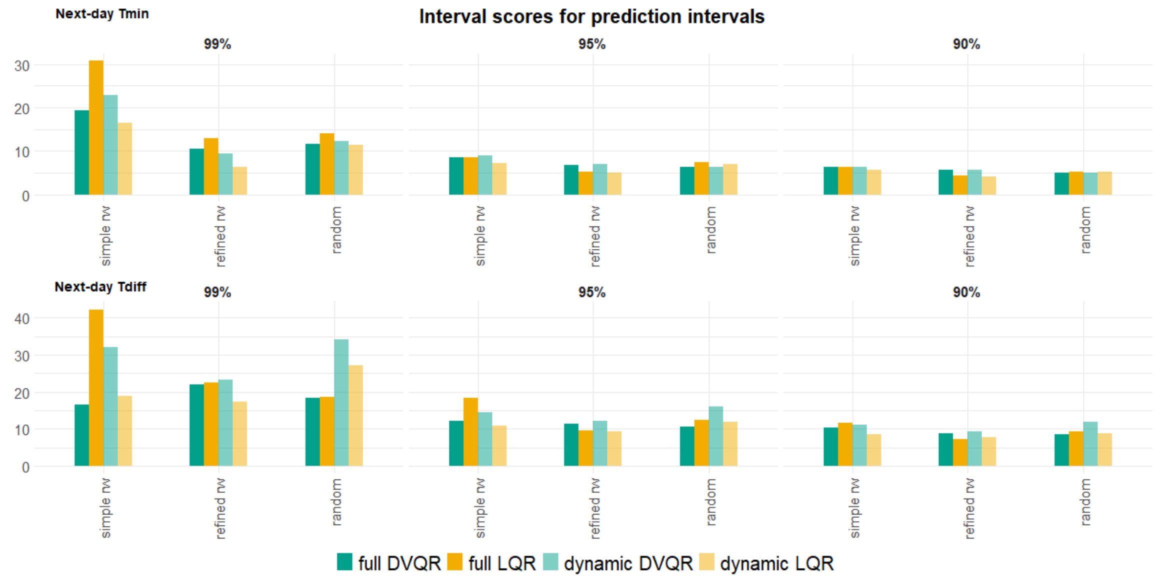


Figure 11: 99%, 95% and 90% prediction interval score for the simple rolling window, refined rolling window and random drawing training approach

look at each training approach and point out a few aspects of the results. For the simple rolling window approach, the dynamic LQR seems to work best since it achieves lowest interval scores for almost all intervals. Interestingly, we observe differences for the two response variables. While for *next-day* T_{min}, the average width of the prediction intervals is the smallest for the dynamic LQR, this is not the case for *next-day* T_{diff}. Instead, the percentages of hits are highest leading to overall low interval scores. On the contrary, the full LQR yields highest interval score values especially for the 99% prediction interval. This is mainly caused by relatively many observations that do not lie in the predicted interval. Unlike the LQR, the DVQR achieves lower interval scores for the full variant where the order of variables is pre-specified. This is also reflected in the percentages of hits. Though, this comes along with slightly broader prediction intervals. In summary, the two LQR variants yield narrowest intervals for *next-day* T_{min}, in return, the intervals obtained from the DVQR methods show a higher percentage of hits. For *next-day* T_{diff}, we observe the inverse. The dynamic DVQR yields narrowest intervals, whereas the dynamic LQR achieves highest relative numbers of hits.

In case of the refined rolling window approach, we observe overall lowest interval score values of all three training approaches. Similar to the simple rolling window, the dynamic LQR performs best in terms of the interval score. Again, we have different results for the two response variables. For *next-day* T_{min}, both variants of the DVQR show very similar results across the three prediction intervals regarding interval score, average width and percentages of hits. For *next-day* T_{diff}, the interval scores also resemble, though the predicted intervals obtained by the dynamic DVQR are narrower, while the full DVQR results in higher percentages of hits. Generally, the two LQR methods yield narrower intervals than the DVQR methods for *next-day* T_{min}, whereas it changes from interval to interval for *next-day* T_{diff}.

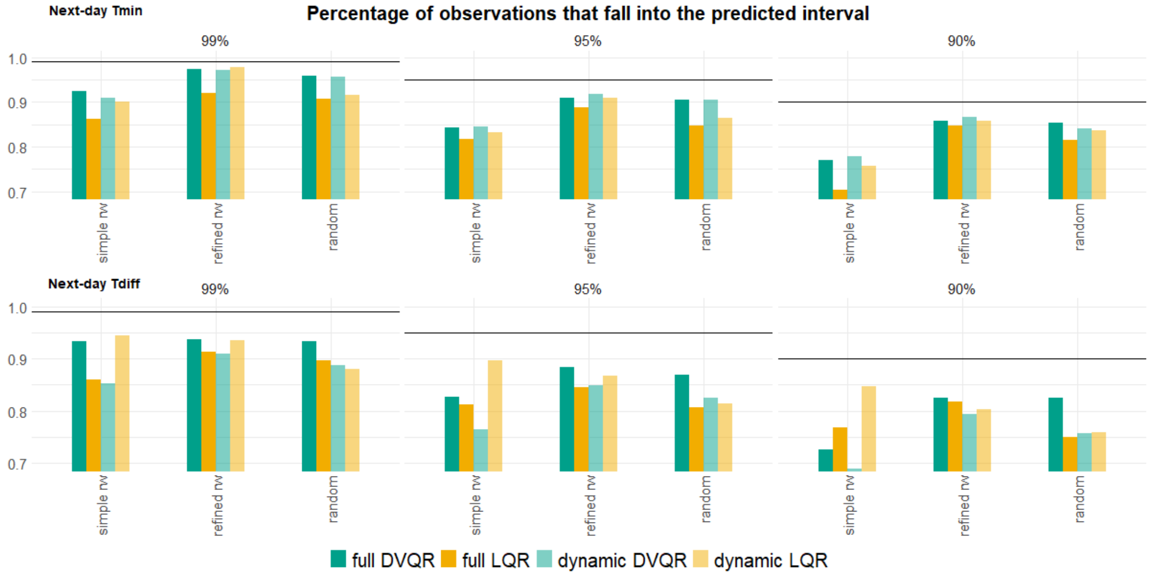


Figure 12: Percentage of observations that fall into the 99%, 95% and 90% prediction interval, for the simple rolling window, refined rolling window and random drawing training approach

For the random drawing approach, we record that the full DVQR without adjusting the order of the D-vine achieves lowest interval scores for all three prediction intervals and both responses. This arises from widest intervals and, for this reason, high percentages of hits. The dynamic DVQR, on the contrary, yields worst interval scores for almost all intervals. Altogether, both DVQR methods outperform the LQR regarding percentages of hits, while it is vice versa with respect to the average width of the intervals.

All in all, we can state some general findings of the plots. For *next-day* T_{min}, the two DVQR variants achieve highest percentages of hits for all three training approaches. This comes at the cost of slightly broader prediction intervals. Regarding differences between the full and dynamic variant of each method, we observe that the full DVQR achieves lower interval score values than the dynamic DVQR. In particular, the full DVQR yields higher percentages of hits whereas the dynamic DVQR leads to narrower prediction intervals. For the LQR, it is nearly the other way round. The dynamic LQR overall bears lower interval scores than the full LQR. More precisely, the dynamic LQR gives higher percentages of hits, while in most cases showing a similar average width of the predicted intervals compared to the full LQR. Lastly, we also recorded from which copula classes the pair copulas in the D-vine were estimated. We witness different shares of copula classes for the two responses. While for *next-day* T_{min}, both elliptical and nonparametric copulas each have a share of approximately 15% of all estimated pair copulas, we observe less elliptical (7%) and more nonparametric (25%) pair copulas for *next-day* T_{diff}. For both responses, between 40% and 50% of the copulas were chosen from the Archimedean class. The remaining pair copulas were estimated as independence copulas. There are no apparent differences among the three different training approaches.

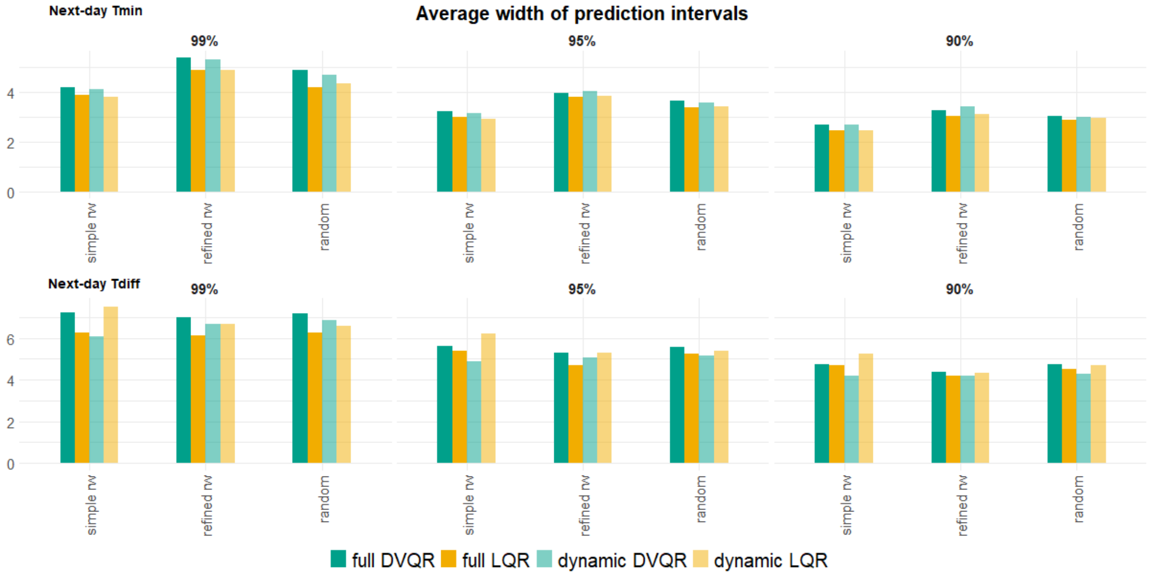


Figure 13: Average width of the 99%, 95% and 90% prediction interval, for the simple rolling window, refined rolling window and random drawing training approach

As a key finding of the shown figures, we conclude that overall, the refined rolling window approach yields best predictive ability. Therefore, we will now concentrate our further examination on this training approach only. First, we take a closer look at the predictions for separate weather stations. This offers some hints to why in many cases the linear quantile regression yields lower interval scores compared to the D-vine regression. Exemplarily, we consider the 95% prediction interval and the scores for the dynamic DVQR and LQR. For the 95% prediction interval of the refined rolling window approach, we do not have any significant differences between the full and dynamic variant anyway. Figure 14 shows for each station, the interval score of the 95% prediction interval. For *next-day* T_{min}, the LQR yields fairly constant interval scores over all 25 stations while the DVQR shows a few spikes, particularly for Station 14 and 15. A reasonable explanation for this behavior might be that the D-vine regression struggles to handle the station-dependent geographical and topological predictor variables. They only take on 25 different values each, which violates the assumption of an i.i.d. sample. It hampers the proper estimation of the marginal distributions. The situation is similar for *next-day* T_{diff}, yet we observe more variability in general.

From our observations so far, we can identify two possibilities to potentially improve predictive ability: Improve fit through a larger training set and reduce the number of predictors to countervail potential overfitting and have less station-specific predictors in the model. In the following we will investigate these two approaches.

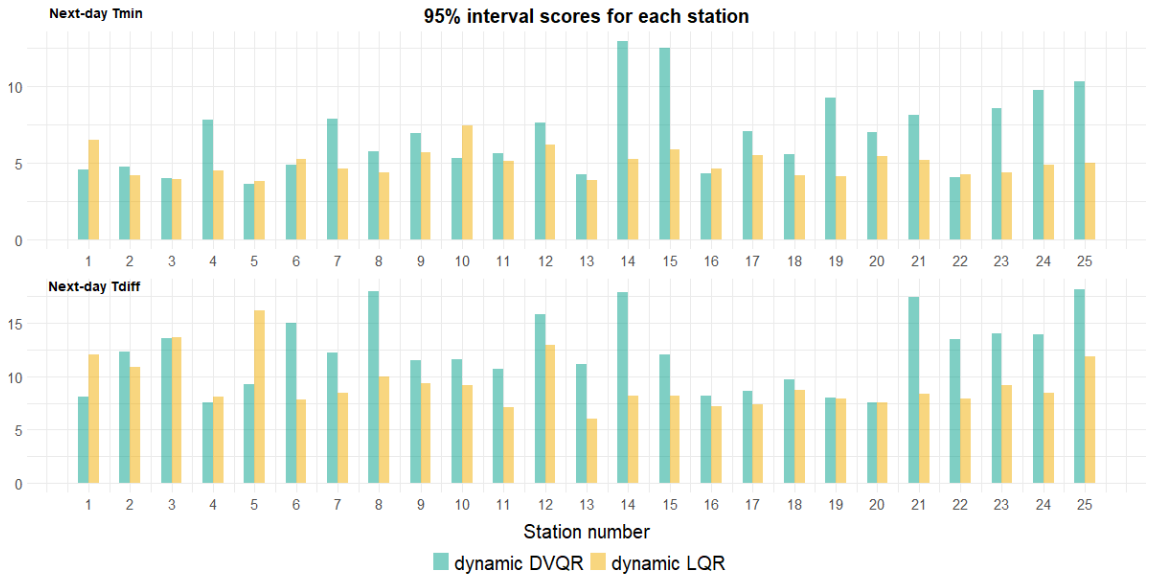


Figure 14: 95% prediction interval score for each station separately, using the refined rolling window approach

4.5.2 Refined Rolling Window Approach with Larger Training Horizon

Since we now solely consider the refined rolling window approach, we can investigate an extension of the training horizon to $n = 7$, i.e. roughly doubling the size of the training set in each instance. Now, the week prior to the predicted day in 2017, as well as two weeks around the predicted day from previous years (2013-2016) enter the training set. This adds up to 67 days compared to 31 days as before. In addition, we can now also train models for more days than before. Due to the training set design, only for the first seven and last seven days of the recorded period in 2017, we cannot train models, still remaining 48 days to predict. First of all, Figure 15 reveals that we obtain overall lower interval scores through the expansion of the training set. Especially, the percentages of hits increased compared to a smaller training horizon (see Figure 16). For *next-day* T_{min}, we could also reduce the average width of the predicted intervals (see Figure 17), while it slightly increased for *next-day* T_{diff}. Compared to a smaller training horizon, the performance of the DVQR particularly improved. Due to the larger number of training samples, the marginal distribution in the DVQR can be estimated more accurately resulting in better predictions (cf. Figure 23 in Appendix A.2 for a comparison of estimated marginals for the smaller and larger training horizon). In the following, we will keep the same training horizon, i.e. using the refined rolling window with $n = 7$. In addition, we will investigate, if we can further improve prediction when reducing the number of predictor variables that enter the model. A lower number of predictor variables is especially desirable for the DVQR in order to reduce complexity and thus runtime of the model fit.

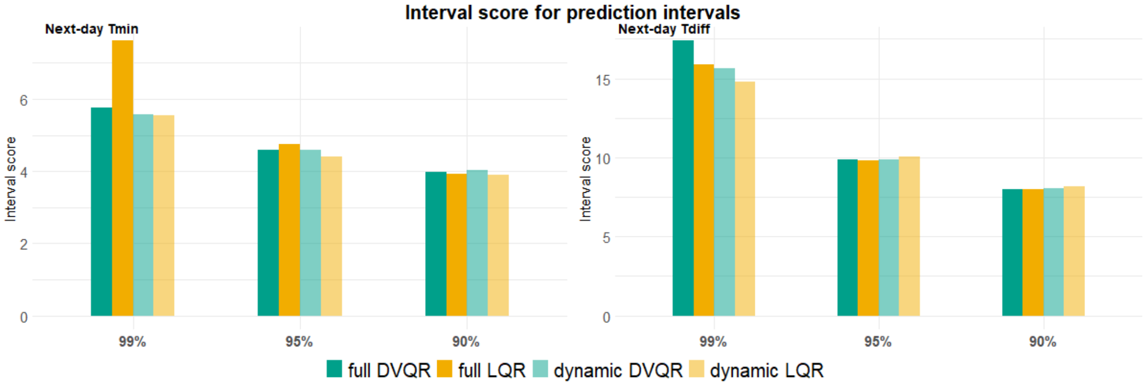


Figure 15: 99%, 95% and 90% prediction interval score for the refined rolling window training approach for $n = 7$

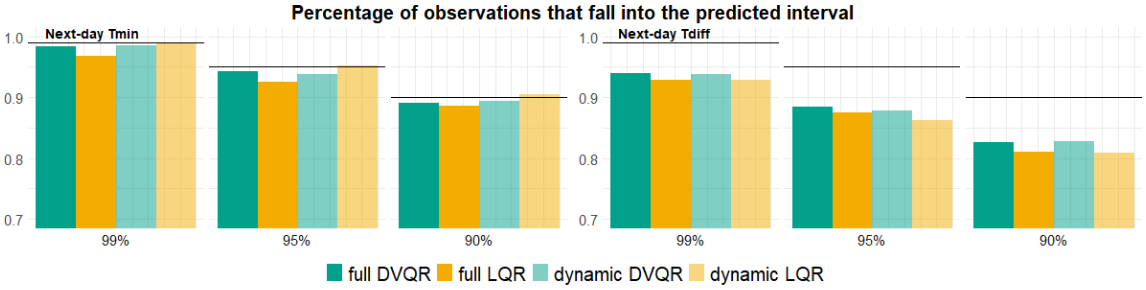


Figure 16: Percentage of observations that fall into the 99%, 95% and 90% prediction interval, for the refined rolling window training approach with $n = 7$

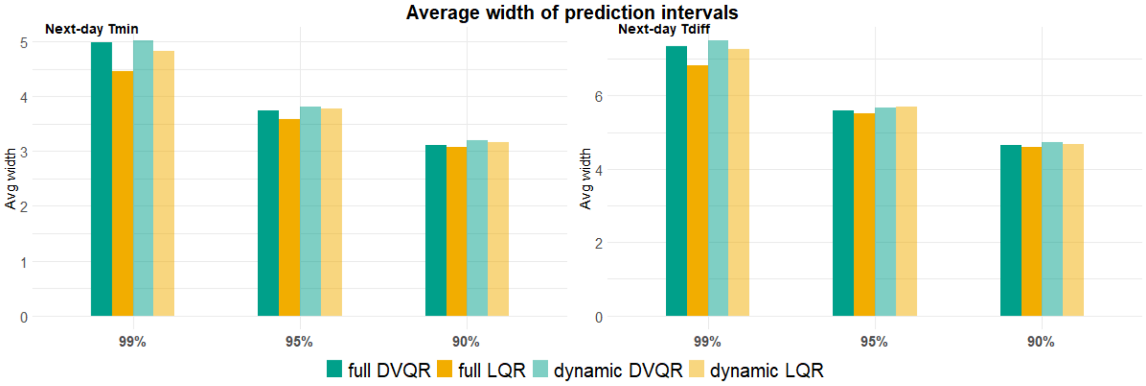
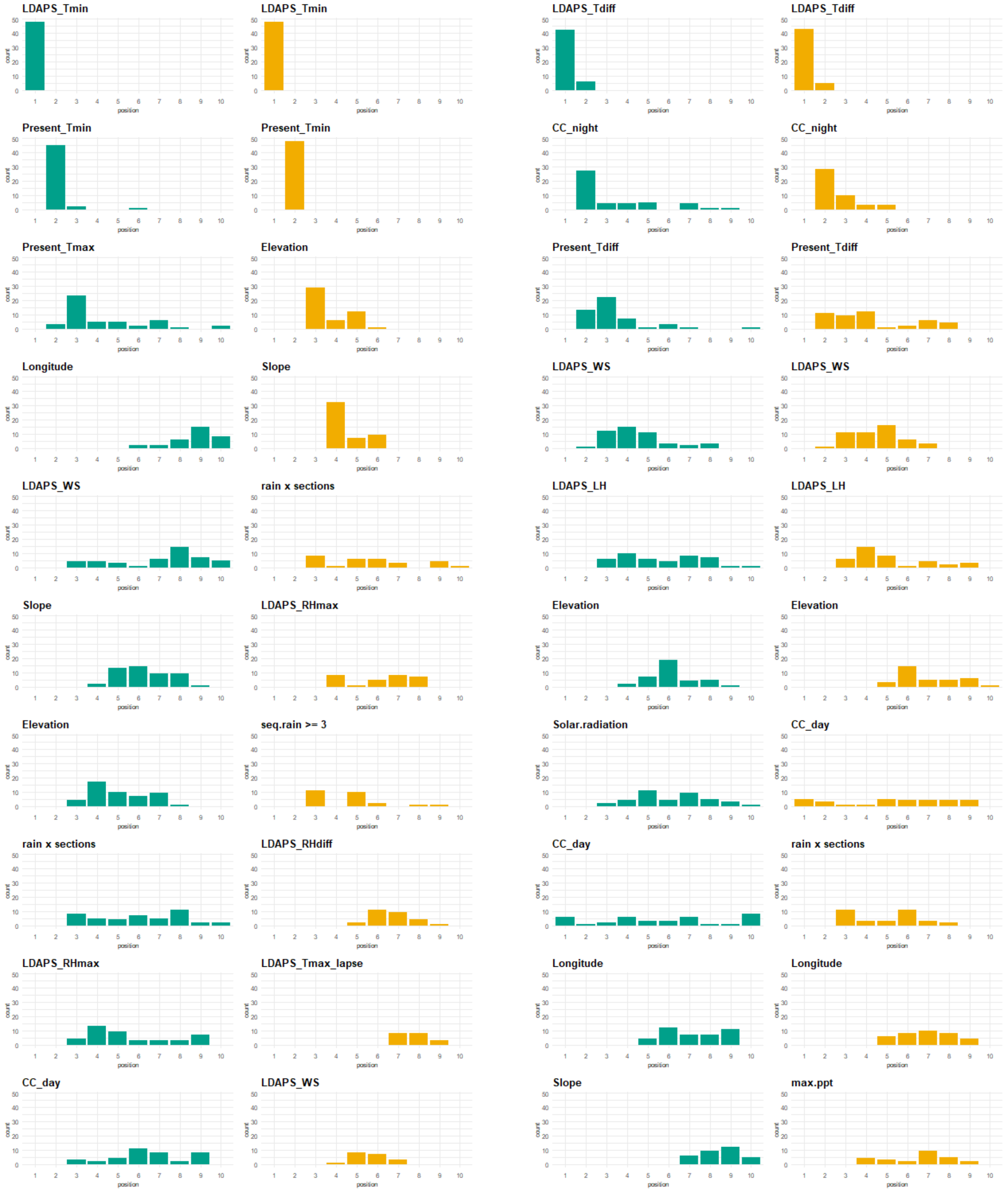


Figure 17: Average width of the 99%, 95% and 90% prediction interval, for the refined rolling window training approach with $n = 7$

4.5.3 Refined Rolling Window Approach with Larger Training Horizon and Less Predictors

When we take a closer look at the respective ordering of the predictors in the dynamic DVQR and LQR model, we, first of all, assert that in most cases not all 10 predictors enter the model.



(a) Predictor selection for *next-day* T_{min}

(b) Predictor selection for *next-day* T_{diff}

Figure 18: Predictor selection for refined rolling window

Hence, the threshold for the number of predictors can justifiably be reduced. Figure 18 displays how often each predictor variable occurs at each position in the D-vine and LQR order, respectively. If, in each training instance, all 10 predictor variables would have been chosen by the DVQR or LQR method, then the sum of frequencies over each position would equal 48, the total number of training instances. This is clearly not the case. The plot reveals another interesting fact. While the first two predictors keep their position in most training instances, we observe some variability in the ordering for the remaining predictors. In many training instances, the order of the predictors deviates from the order that was identified in the model selection process when we fitted a model to the whole data from 2013-2016 (cf. Section 4.3). This particularly applies to the predictor *longitude* in the DVQR model for *next-day* T_{min}. While it appeared at the fourth position in the ordering when fitting a model on 2013-2016, it does not appear at any front position now. Furthermore, for *next-day* T_{diff}, the predictor CC_{day} took almost every position in the ordering of the DVQR and LQR, even first position in a few training instances.

order	DVQR	LQR
1	LDAPS $\mathrm{T_{min}}$	LDAPS $\mathrm{T_{min}}$
2	Present-day $\mathrm{T_{min}}$	Present-day $\mathrm{T_{min}}$
3	Present-day $\mathrm{T_{max}}$	Elevation
4	Elevation	Slope
5	LDAPS $\mathrm{RH_{max}}$	$\mathrm{seq.rain_{\geq 3}}$

(a) Response: *next-day* T_{min}

order	DVQR	LQR
1	LDAPS $\mathrm{T_{diff}}$	LDAPS $\mathrm{T_{diff}}$
2	$\mathrm{CC_{night}}$	$\mathrm{CC_{night}}$
3	Present-day $\mathrm{T_{diff}}$	Present-day $\mathrm{T_{diff}}$
4	LDAPS WS	LDAPS WS
5	$\mathrm{CC_{day}}$	LDAPS LH

(b) Response: *next-day* T_{diff}

Table 3: Selected predictors in the reduced models

These findings justify a more restrictive threshold on the number of predictors. We will now train models for each day again, allowing for five predictor variables only. Therefore, for each response and method, we want to select those five predictors that most often appeared at one of the first five positions in the ordering. We add up the occurrences of each predictor variable at one of the first five positions, giving ascending weight to the positions. For example, let X be one of the predictor variables and $pos_X = (22, 13, 8, 4, 1)^T$ its number of occurrences at each of the first five positions in the ordering. Then, we multiply pos_X by the weight vector $w = (5, 4, 3, 2, 1)^T$ resulting in a score

$$s_X := w^T pos_X$$

for each predictor variable. For each response variable and method, we rank the predictor variables by this score giving us the orderings in Table 3. We will use the same enlarged training horizon for the refined rolling window approach as before ($n = 7$ as in Section 4.5.2). In each instance, we have a training set size of 67 days. In total, 48 days will be predicted. The resulting interval scores are shown in Figure 19, the percentages of hits in

Figure 20 and the average width of the predicted intervals in Figure 21.

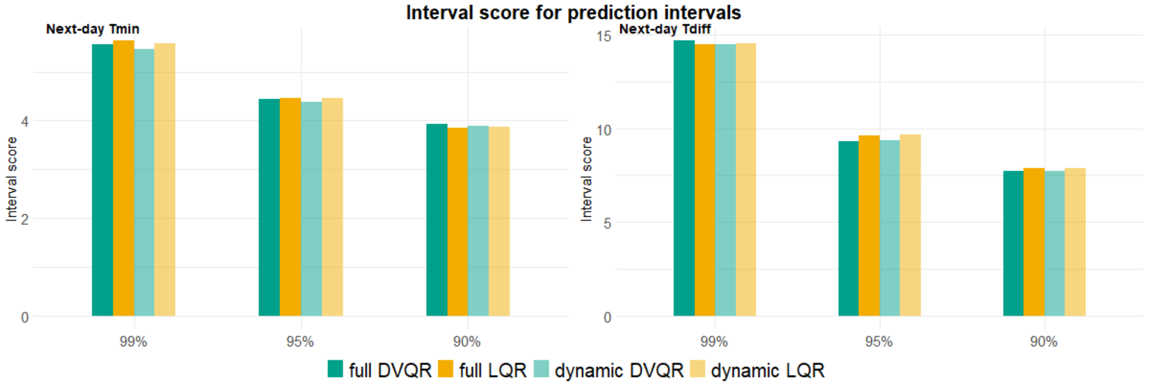


Figure 19: 99%, 95% and 90% prediction interval score for the refined rolling window training approach for $n = 7$ and five predictor variables

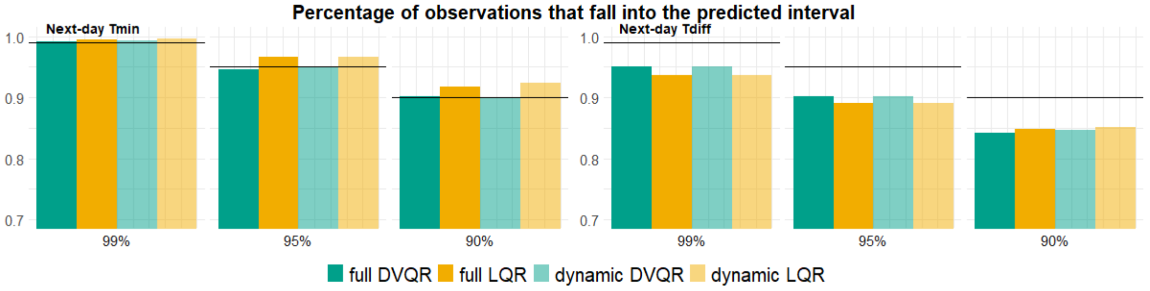


Figure 20: Percentage of observations that fall into the 99%, 95% and 90% prediction interval, for the refined rolling window training approach with $n = 7$ and five predictor variables

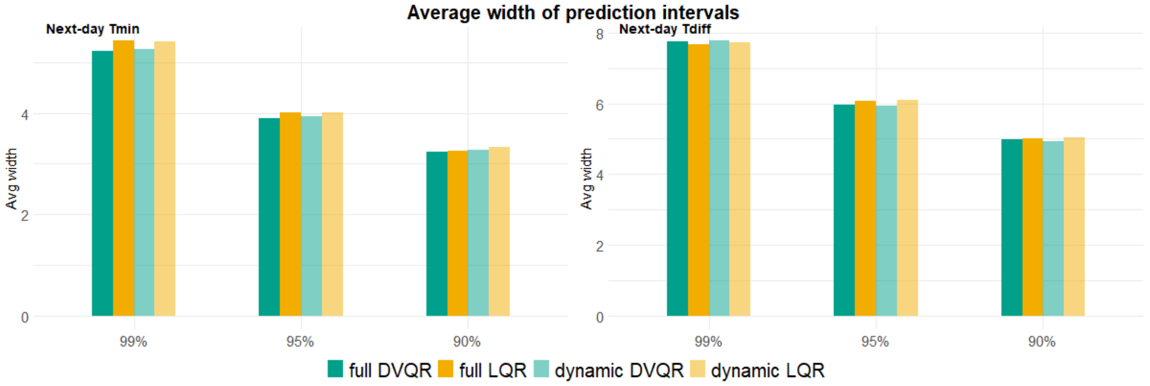


Figure 21: Average width of the 99%, 95% and 90% prediction interval, for the refined rolling window training approach with $n = 7$ and five predictor variables

First and foremost, we could rigorously reduce runtime using only five predictor variables in the model. Compared to the models with 10 predictor variables, we even observe a little improvement of the interval score values. This is mainly due to a larger number of observations that fall into the predicted intervals. The average width of the prediction intervals slightly increase compared to the models with 10 predictor variables. Altogether, we now have less variability between the different regression methods and variants. Remarkably, the dynamic DVQR yields lowest interval scores for all intervals, except for the 90% prediction interval for *next-day* T_{min}. The greater improvement of the DVQR compared to the LQR, could be due to the absence of most topological and geographical predictors. In the reduced model setting with five predictors, any of these variables appears in the D-vine for *next-day* T_{diff} and *Elevation* is the only one that was selected for *next-day* T_{min}. The differences between the full and dynamic variant of the regression methods are marginal. In Appendix A.2, the estimated 95% prediction intervals of the dynamic DVQR and LQR are explicitly shown, separately for each response variable and station. At last, we look at the copula classes from which the pair copulas in the D-vine were estimated. Figure 22 shows the shares of each copula class, separately for each training instance and in total. As we have mentioned before, the share of elliptical copulas is larger for *next-day* T_{min} than for *next-day* T_{diff}, while for *next-day* T_{diff} more pair copulas were estimated as nonparametric copulas. In most training instances, the majority of pair copulas were estimated from the Archimedean copula class thus accounting for asymmetries and tail dependencies between the variables.

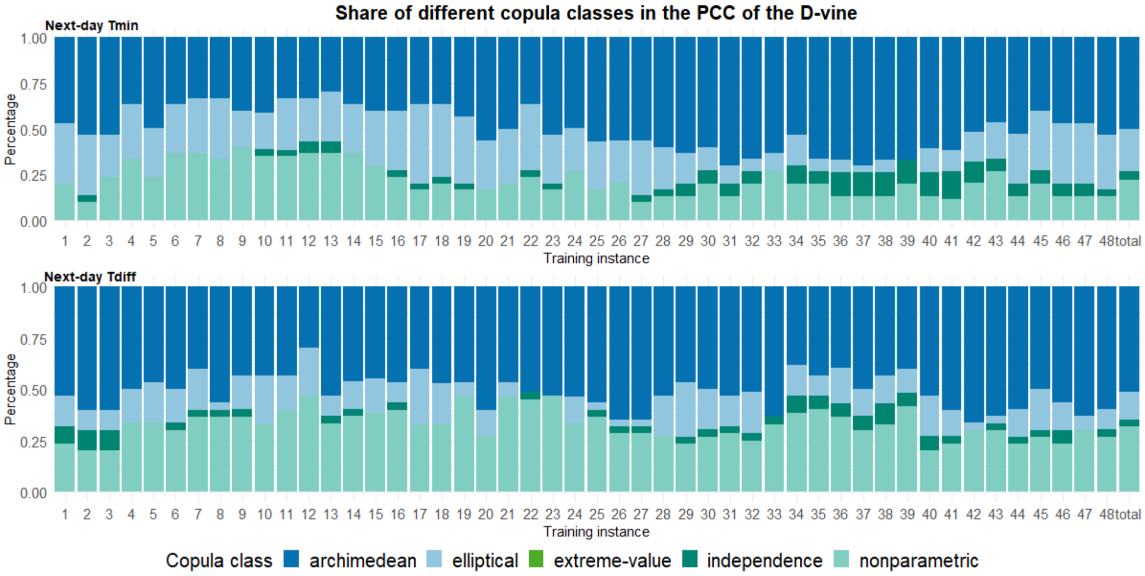


Figure 22: Share of different copula classes in the D-vine of each training instance

4.5.4 Notes on Runtime

Regarding runtime of the two methods, the DVQR took considerably longer to fit a model than the LQR. For example, the runtime for fitting a LQR model with 10 predictor variables to a training instance containing approximately 1,600 observations lied within a few seconds. On the contrary, fitting a DVQR model to the same data took several minutes. Expectedly, the DVQR with a fixed ordering of the predictors and the DVQR allowing for an automated forward selection of the predictors differed noticeably in runtime. While, on average, it took 4 min to fit a DVQR model with a fixed ordering of the 10 predictors, allowing for a dynamic selection of the predictors resulted in a mean runtime of 15 min per training instance. Reducing the number of predictors included in the model to five, could clearly decrease runtime of the DVQR. Now, fitting a DVQR model with fixed ordering took approximately 1 min per training instance, a dynamic DVQR model roughly 2.5 min. In particular, the estimation of the pair copulas for the D-vine density is costly and the number of pair copulas that need to be estimated obviously depends on the number of predictors: quadratically in case of a DVQR model with fixed ordering of the predictors and cubically for a dynamic DVQR model. More precisely, given d predictors, a DVQR model with fixed ordering needs to estimate

$$\sum_{i=1}^{d} i = d(d+1)/2$$

pair copulas. In the first tree of the D-vine we have d pair copulas, in the second tree $d-1$, and so on. When we allow for an automated predictor selection, in each step, the DVQR method needs to identify from all remaining predictors the one that would most increase the model's fit. Therefore, in step k, this requires the estimation of k pair copulas for all predictors not yet in the model (cf. Figure 3 for an illustration). Hence, the total number of estimated pair copulas adds up to

$$\sum_{i=1}^{d} i(d+1-i) = d(d+1)(d+2)/6.$$

The relationship between the number of predictors and the runtime conforms to our recorded runtimes. Another factor that decisively controls runtime of the DVQR model fit is the variety of copula classes from which the DVQR can choose. In our application, we did not set any restrictions, thus allowing for all possible copula classes that are implemented in the `VineCopula` package. Additionally, we included nonparametric copulas. Of course, the larger the number of possible copula families, the longer the estimation of the pair copulas takes. Therefore, in practical applications, one could reduce runtime by trying to isolate a small number of appropriate copula families for the dependence structure of specific variable pairs. This could also be done in a pre-processing step.

5 Conclusion

In this master's thesis, we compared predictive ability of the D-vine copula-based and linear quantile regression method on sizeable real data. Therefore, we revisited the underlying theory in the first part of the thesis. The data we have chosen for this application originates from the UCI machine learning repository and contains numerical weather data that was recorded in Seoul, South Korea. In our application, we aimed at estimating 99%, 95% and 90% prediction intervals for next-day minimum and difference temperature values using predicted conditional quantiles. We evaluated predictive ability quantified by the interval scoring and the check loss. Further, we looked at the relative number of observations that indeed fell into the predicted interval as well as the average width of the intervals. In a first step, we investigated three different approaches for a training set selection: a simple rolling window, a refined rolling window and a random training set composition. To compare these three approaches, we used a training horizon of 31 days in each case. From these three approaches, we identified the refined rolling window to work best with respect to predictive ability, though observing slightly better interval score values for the LQR method. Focussing solely on the refined rolling window approach, we roughly doubled the size of the training horizon in a second step. Expectedly, predictive ability increased for both methods, yet the DVQR achieved comparatively greater improvement. We observed that for most training instances, the BIC-based automated forward-selection of predictor variables eliminated a few predictors from the model. Thus, we set a restrictive threshold of five predictor variables in a third step. This threshold could increase the relative number of observations in the predicted intervals while reasonably, the average width of the intervals also slightly increased by 7% on average. Still, the interval scores could be further reduced. Regarding the estimated pair copulas in the D-vine density, we observed that for both response variables half of the pair copulas were estimated as archimedean copulas. For *next-day* T_{min}, approximately 25% of all pair copulas were estimated as elliptical copulas and 20% as nonparametric. For *next-day* T_{diff}, we witnessed a shift towards nonparametric copulas now amounting to 35% of all estimated pair copulas while only 10% of the copulas were selected from the elliptical family. This clearly implies that using Gaussian copulas only, would not provide appropriate model fits. Yet, the benefits of flexibly modelling the dependence structures in the DVQR come at the cost of considerably longer runtimes compared to the LQR.

To further improve predictive ability one could consider previous temperature values of the response variables as predictors, i.e. incorporating autoregressive models to account for potential time series effects. Marginal time series structures might also be present for the time-dependent predictor variables. One approach to handle this would be fitting autoregressive models to each of the margins and using standardized residuals to transform to pseudo-copula data. The standardized residuals are then approximately i.i.d. data. Another interesting consideration would be a forecast of several days in advance. While we focussed on one-step-ahead predictions, one could also study n-steps-ahead predictions. Though, this would also require n-steps-ahead predictions of the LDAPS model in real-time applications. Another open research topic, for which this application and data could serve as a starting point, is the simultaneous modelling of two response variables in the D-vine quantile regression.

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics 44*(2), 182–198.

Abrevaya, J. (2002). *The effects of demographics and maternal behavior on the distribution of birth outcomes*, pp. 247–257. Heidelberg: Physica-Verlag HD.

Bacmeister, J. T. (2012). *Weather Prediction Models*, pp. 89–114. Springer New York.

Bedford, T. and R. M. Cooke (2002). Vines - a new graphical model for dependent random variables. *The Annals of Statistics 30*, 1031–1068.

Bernard, C. and C. Czado (2015). Conditional quantiles and tail dependence. *Journal of Multivariate Analysis 138*, 104–126.

Cho, D., C. Yoo, J. Im, and D.-H. Cha (2020). Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas. *Earth and Space Science 7*.

Czado, C. (2019). *Analyzing dependent data with vine copulas.* Springer.

Dodge, Y. (2008). *Regression Analysis*, pp. 450–452. New York, NY: Springer New York.

Duong, T. (2016). Non-parametric smoothed estimation of multivariate cumulative distribution and survival functions, and receiver operating characteristic curves. *Journal of the Korean Statistical Society 45*, 33–50.

Gneiting, T. and A. E. Raftery (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association 102*(477), 359–378.

Groß, J. (2003). *Lecture Notes in Statistics: Linear Regression.* Springer Berlin Heidelberg.

Handcock, M. S. and M. Morris (1999). *Quantile Regression, In: Relative Distribution Methods in the Social Sciences.* Springer New York.

Hui-Kuang Yu, T. (2014). A quantile regression forecasting model for ict development. *Management Decision 52*, 1263–1272.

Joe, H. (1997). *Multivariate models and multivariate dependence concepts.* Chapman and Hall.

Koenker, R. (2005). *Quantile Regression.* Econometric Society Monographs. Cambridge University Press.

Koenker, R. (2013). *quantreg: Quantile regression.* R package version 5.74.

Koenker, R. and G. J. Bassett (1978). Regression quantiles. *Econometrica 46*, 33–50.

Koenker, R. and K. F. Hallock (2001). Quantile regression. *Journal of Economic Perspectives 15*, 143–156.

Kraus, D. and C. Czado (2017). D-vine copula based quantile regression. *Computational Statistics and Data Analysis 110*, 1–18.

Latif, S. and F. Mustafa (2020). Parametric vine copula construction for flood analysis for kelantan river basin in malaysia. *Civil Engineering Journal 6*(8), 1470–1491.

Liao, R., P. Boonyakunakorn, J. Liu, and S. Sriboonchitta (2019, Oct). Modelling dependency structures of crude oil prices and stock markets of developed and developing countries: A c-vine copula approach. *Journal of Physics: Conference Series 1324*, 012097.

Loader, C. (1999). *Local Regression and Likelihood*. Springer.

Lynch, P. (2006). *The Emergence of Numerical Weather Prediction: Richardson's Dream*. Cambridge University Press.

Möller, A., L. Spazzini, D. Kraus, T. Nagler, and C. Czado (2018). Vine copula based post-processing of ensemble forecasts for temperature.

Nagler, T. (2016a, 03). kdecopula: An r package for the kernel estimation of copula densities. *Journal of Statistical Software 84*.

Nagler, T. (2016b). *kdecopula: Kernel Smoothing for Bivariate Copula Densities*. R package version 0.9.2.

Nagler, T., C. Bumann, and C. Czado (2019). Model selection in sparse high-dimensional vine copula models with an application to portfolio risk. *Journal of Multivariate Analysis 172*, 180–192. Dependence Models.

Nagler, T. and D. Kraus (2020). *vinereg: D-Vine Quantile Regression*. R package version 0.7.2.

Nagler, T., C. Schellhase, and C. Czado (2017). Nonparametric estimation of simplified vine copula models: comparison of methods. *Dependene Modeling 5*, 99–120.

Nagler, T., U. Schepsmeier, J. Stoeber, E. C. Brechmann, B. Graeler, and T. Erhardt (2020). *VineCopula: Statistical Inference of Vine Copulas*. R package version 2.4.1.

Nelson, R. B. (2006). *An Introduction to Copulas*. Springer.

Noh, H., A. E. Ghouch, and I. V. Keilegom (2015). Semiparametric conditional quantile estimation through copula-based multivariate models. *Journal of Business & Economic Statistics 33*(2), 167–178.

Olive, D. J. (2017). *Linear Regression*. Springer.

Panagiotelis, A., C. Czado, and H. Joe (2012). Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association 107*(499), 1063–1072.

Schallhorn, N., D. Kraus, T. Nagler, and C. Czado (2017). D-vine quantile regression with discrete variables.

Sen, A. and M. Srivastava (1990). *Regression Analysis.* Springer.

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l´Institut Statistique de l´Université de Paris 8*, 229–231.

Spokoiny, V., W. Wang, and W. K. Härdle (2013). Local quantile regression. *Journal of Statistical Planning and Inference 143*(7), 1109–1129.

Tepegjozova, M. (2019, Nov). D- and c-vine quantile regression for large data sets. Masterthesis, Technische Universität München, Garching b. München.

Weiss, N., P. Holmes, and M. Hardy (2005). *A Course in Probability.* Pearson Addison Wesley.

Yu, K., Z. Lu, and J. Stander (2003). Quantile regression: applications and current research areas. *The Statistician 52*, 331–350.

A Further Results and Plots

A.1 Selected Predictors in the Model Selection Process

pool of predictors	Position	
	DVQR	LQR
LDAPS T_{min}	1	1
LDAPS T_{max}	12	9
Present-day T_{min}	2	2
Present-day T_{max}	3	-
LDAPS RH_{max}	9	6
LDAPS RH_{diff}	11	8
LDAPS WS	5	10
LDAPS LH	13	-
Solar radiation	18	-
CC_{day}	10	-
CC_{night}	15	-
Elevation	7	3
Slope	6	4
Latitude	17	-
Longitude	4	11
seq.rain$_{\geq 3}$	-	7
rain × sections	8	5
PPT_{max}	16	-
PPT_{avg}	19	-
$PPT_{18\text{-}23\text{ h}}$	14	-

(a) Response: *next-day T_{min}*

pool of predictors	Position	
	DVQR	LQR
LDAPS T_{diff}	1	1
LDAPS T_{max}	-	-
Present-day T_{min}	-	13
Present-day T_{diff}	3	3
LDAPS RH_{min}	-	-
LDAPS RH_{diff}	11	-
LDAPS WS	4	4
LDAPS LH	5	5
Solar radiation	7	11
CC_{day}	8	7
CC_{night}	2	2
Elevation	6	6
Slope	10	-
Latitude	-	-
Longitude	9	9
seq.rain$_{\geq 3}$	-	12
rain × sections	13	8
PPT_{max}	14	10
PPT_{avg}	-	-
$PPT_{6\text{-}11\text{ h}}$	12	-

(b) Response: *next-day T_{diff}*

Table 4: Set of eligible and selected predictors for each model type and response

A.2 Marginal Fits, Prediction Intervals and Check Loss

For the DVQR, Figure 23 compares the marginal fits of the predictors in the reduced model setting (cf. Section 4.5.3) and the two response variables for the smaller and larger training horizon. The figure exemplarily shows the marginal fits of one training instance. The histograms represent the whole data from 2013-2017 in order to tell how well the estimated marginals fit the ”true” distribution of the particular variables. As expected, the larger training horizon leads to a more appropriate fit of the marginals.

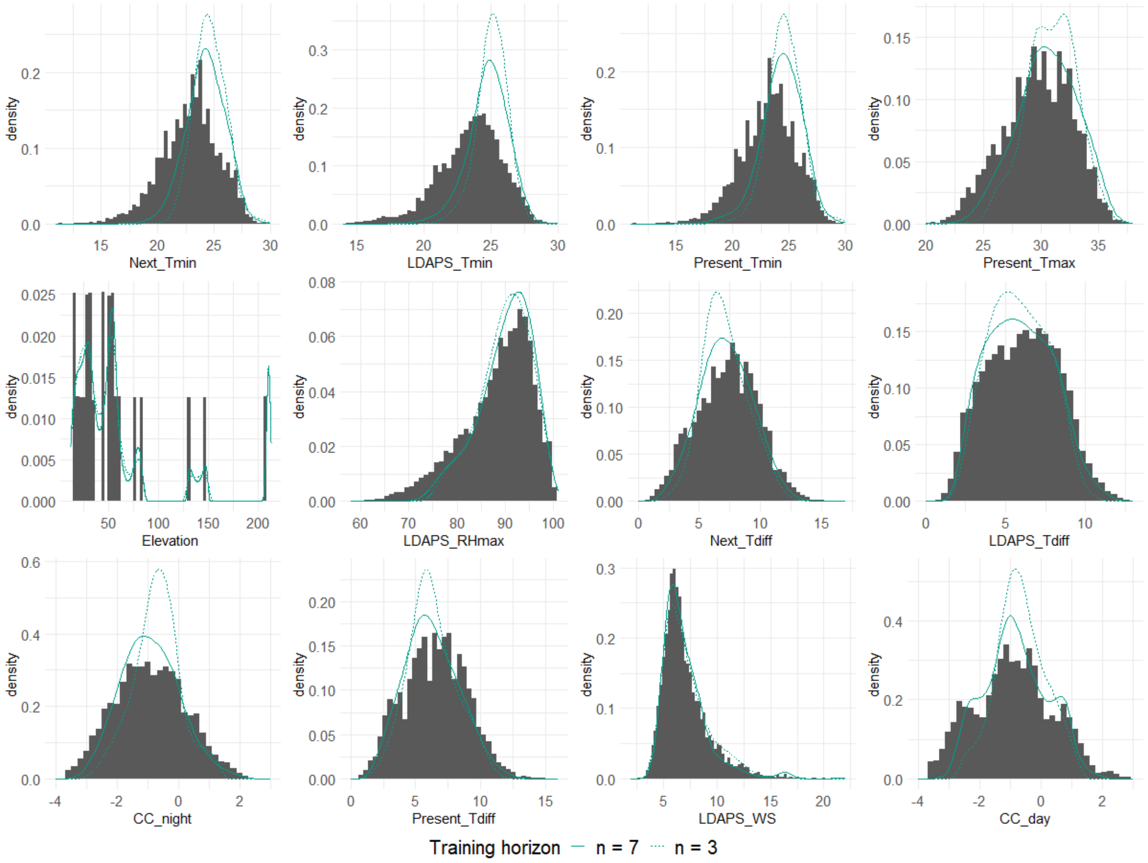


Figure 23: Comparison of marginal fits for the smaller and larger training horizon of the refined rolling window approach

Figure 24 - 26 show the estimated 95% prediction intervals for the refined rolling window training approach with $n = 7$ and five predictor variables (cf. Section 4.5.3). The left column shows the predicted intervals for the response *next-day* T_{min} and the right column for *next-day* T_{diff}. For *next-day* T_{min}, most of the predicted intervals are very similar for the DVQR and LQR method. For the 35th predicted day, which corresponds to August 10, the DVQR predicts a remarkably large interval across all weather stations. Especially the lower end of the interval is much smaller than for the LQR. Still, for all 25 stations, the true minimum temperature lies within the predicted interval. Overall, the results differ for each station. For some stations, e.g. station 7, the LQR usually yields smaller values for the lower endpoints of the predicted intervals while the DVQR yields larger values for the higher endpoints of the intervals. For other stations it is vice versa, e.g.

station 4. Also the position of the true temperature value within the intervals changes from station to station. While in many instances, the true temperature value lies nearly in the middle of the predicted interval, we can observe a shift towards the upper or lower end of the intervals for some stations. For *next-day* T_{diff}, comparatively more observations do not lie in the predicted intervals. Also, for predicted intervals that do not include the true temperature value, the true value lies relatively far from the interval in some cases. This both reflects the higher interval scores for *next-day* T_{diff} compared to *next-day* T_{min}. Differences between the predicted intervals of the DVQR and LQR method are slightly more present than for *next-day* T_{min}.

Figure 27 compares the check loss at nine different alpha levels of all three training approaches with a fixed training horizon of 31 days. For most training approaches and alpha levels, the LQR yields lower check loss values. Figure 28 shows the check loss at the different alpha levels for the refined rolling window with a larger training horizon of 67 days. Here, we observe very similar results for the DVQR and LQR method. While the DVQR achieves slightly lower check loss values for larger quantiles, the LQR outperforms the DVQR for smaller quantiles. Lastly, Figure 29 displays the check loss at the different alpha levels for the refined rolling window with a larger training horizon of 67 days and a threshold of five on the number of predictors in the model. For *next-day* T_{min}, the DVQR again obtains lower values for larger quantiles while it is vice versa for the 0.05, 0.25 and 0.5 quantile. The check loss for the 0.005 and 0.025 quantile hardly differ. For *next-day* T_{diff}, the LQR also yields lower check loss values for quantiles towards the center of the distribution (i.e. the 0.25, 0.5 and 0.75 quantile) while the DVQR achieves lower values towards the tails of the distribution. This conforms to our expectation since the DVQR method is able to particularly capture tail dependencies due to its design.

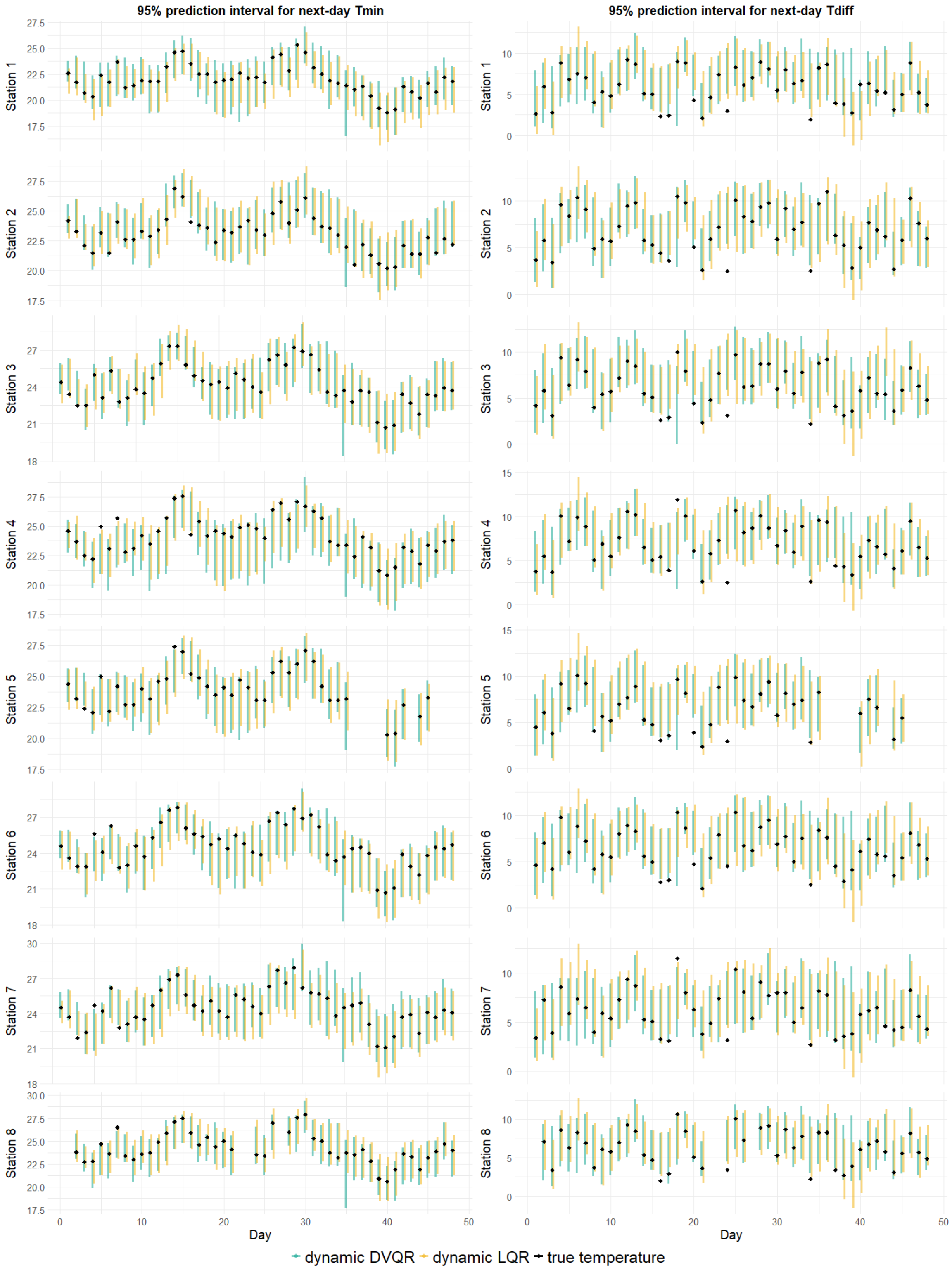


Figure 24: Predicted 95% intervals and actual temperature values for the refined rolling window training approach with $n = 7$ and five predictor variables — Stations 1-8

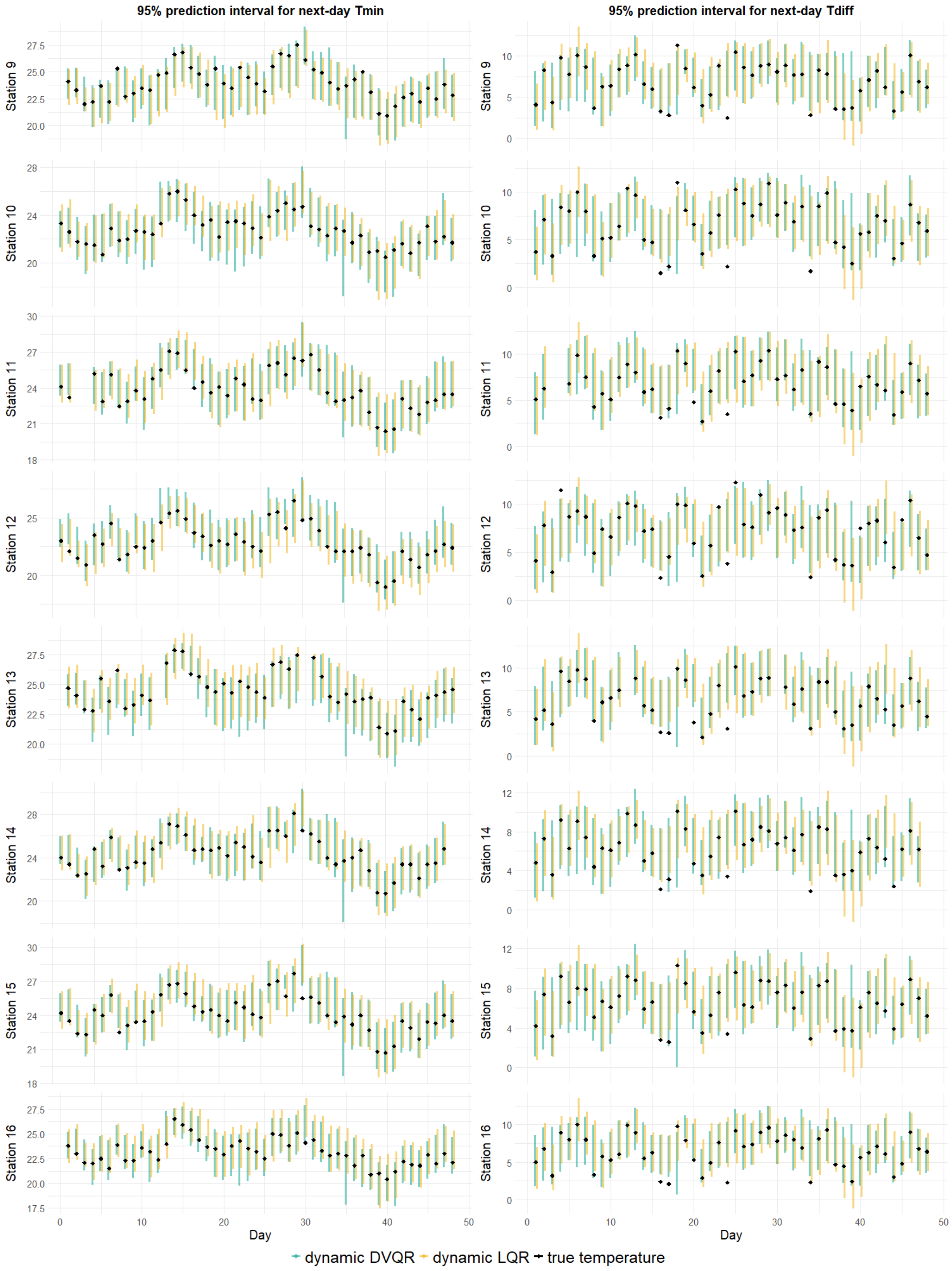


Figure 25: Predicted 95% intervals and actual temperature values for the refined rolling window training approach with $n = 7$ and five predictor variables — Stations 9-16

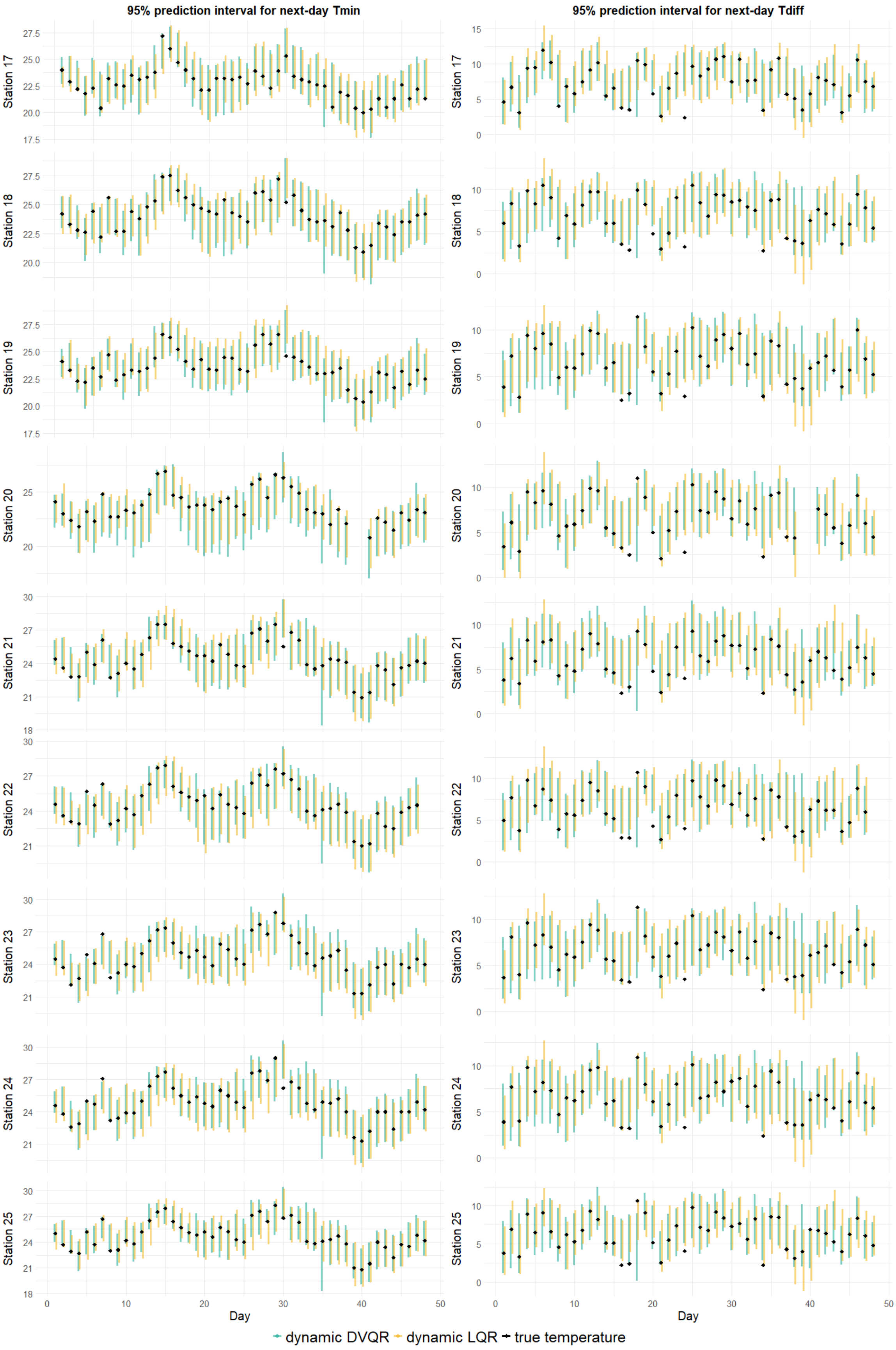


Figure 26: Predicted 95% intervals and actual temperature values for the refined rolling window training approach with $n = 7$ and five predictor variables — Stations 17-25

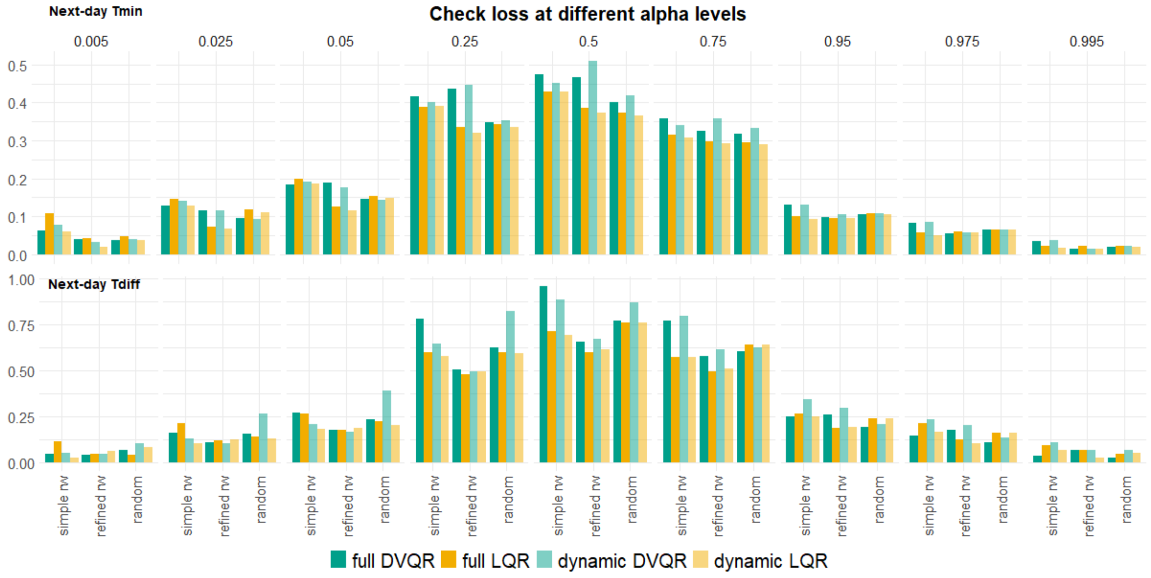


Figure 27: Check loss of next-day predictions at alpha levels 0.005, 0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975 and 0.995, for the simple rolling window, refined rolling window and random drawing training approach

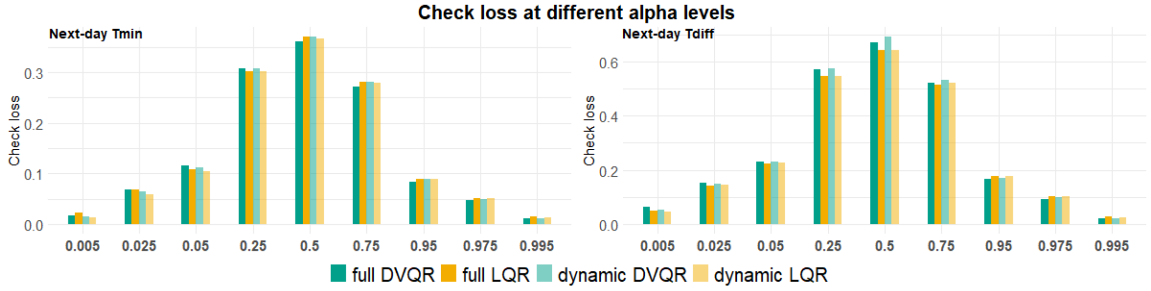


Figure 28: Check loss of next-day predictions at alpha levels 0.005, 0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975 and 0.995, for the refined rolling window training approach with $n = 7$ and 10 predictor variables

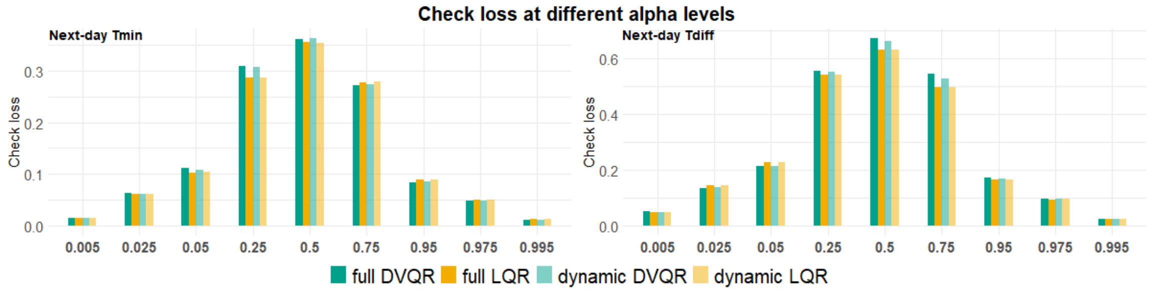


Figure 29: Check loss of next-day predictions at alpha levels 0.005, 0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975 and 0.995, for the refined rolling window training approach with $n = 7$ and five predictor variables