Technical Report

UCAM-CL-TR-676
ISSN 1476-2986

Number 676

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Road traffic analysis using MIDAS data: journey time prediction

R.J. Gibbens, Y. Saacti

December 2006

Department for Transport Horizons Research Programme "Investigating the handling of large transport related datasets" (project number H05-217)

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Executive summary

The project described in this report was undertaken within the Department for Transport's second call for proposals in the Horizons research programme under the theme of *Investigating the handling of large transport related datasets*. The one year project started on 1 October 2005 with funding for Yunus Saatci as a post-graduate research associate and with Richard Gibbens as the principal investigator.

In this report we describe our findings from a project using the combination of historical and real-time MIDAS loop detector data for journey time prediction. Phase one of the project involved a short study of the data formats used by MIDAS to record traffic count data and described a revised data format including explicit indexing. This revised format, based on the familiar ZIP file archiving tool allowed efficient random access to the data necessary for high throughput applications.

The project looked at the variability of journey times across days in three day categories: Mondays, midweek days and Fridays. Two estimators using real-time data were considered: a simple-to-implement regression-based method and a more computationally demanding k-nearest neighbour method. Our example scenario of UK data was taken from the M25 London orbital motorway during 2003 and the results compared in terms of the root-mean-square prediction error. It was found that where the variability was greatest (typically during the rush hours periods or periods of flow breakdowns) the regression and nearest neighbour estimators reduced the prediction error substantially compared with a naïve estimator constructed from the historical mean journey time. Only as the lag between the decision time and the journey start time increased to beyond around 2 hours did the potential to improve upon the historical mean estimator diminish. Thus, there is considerable scope for prediction methods combined with access to real-time data to improve the accuracy in journey time estimates. In so doing, they reduce the uncertainty in estimating the generalized cost of travel. The regression-based prediction estimator has a particularly low computational overhead, in contrast to the nearest neighbour estimator, which makes it entirely suitable for an online implementation.

Finally, the project demonstrates both the value of preserving historical archives of transport related datasets as well as provision of access to real-time measurements.

Contents

List of Figures

1 Introduction

The project described in this report was undertaken within the Department for Transport's second call for proposals in the Horizons research programme under the theme of *Investigating the handling of large transport related datasets*. The one year project started on 1 October 2005 with funding for Yunus Saatci as a post-graduate research associate and with Richard Gibbens as the principal investigator.

Phase one of the project reported on issues of MIDAS data organization [1] and phase two [2] presented interim findings on journey time prediction methodolgies. This report gathers together these interim reports into a final form.

Journey time prediction using sources of real-time measurement data has the potential to assist travellers through the provision of more accurate estimates of journey times. Improving the accuracy of the prediction by suitable methods that make use of real-time data helps to reduce the overall uncertainty of journey times.

Rice & van Zwet [5] describe a simple-to-implement prediction methodology and report successful results with US data in comparison with more sophisticated and harder-to-implement methods. In this project we have examined in detail the performance of these methodologies when used with real-time UK MIDAS loop detector data. A preliminary account of this investigation is given in [3, 6].

The work on data organization carried out in phase one of the project provided the essential underpinning for our numerical investigations that followed and is briefly summaried here as Appendix A.

Section 2 describes the basic model and defines the prediction methodologies considered. Section 3 presents the results of our numerical investigations into journey times and the comparison between the methodologies. Conclusions are given in Section 4. An executive summary is also included.

2 Journey time prediction methodologies

2.1 Basic model and notation

The basic model and terminology are taken directly from Rice & van Zwet [5] and are briefly summarized here as follows.

We suppose that there is a *velocity field*, $V(d, \ell, t)$, specifying the average speeds of vehicles for days $d \in D$, at loop detectors, $\ell \in \{1, \ldots, L\}$ and for times (of day) $t \in T$. There may be many days d and journeys are traversed from loop 1 to loop L. The time of day epochs, t, are taken as every minute in the case of MIDAS data.

Define $T(d, t)$ for the time of travel from loop 1 to loop L starting at time t on day d. $T(d, t)$ can be determined (approximately) from the velocity field on any day d in the past.

Define also, a *frozen-field* travel time, $T^*(d,t)$, given by

$$T^*(d,t) = \sum_{\ell=1}^{L-1} \frac{2d_\ell}{V(d,\ell,t) + V(d,\ell+1,t)} \tag{1}$$

where d_ℓ is the distance between loops ℓ and $\ell+1$. This quantity will play a pivotal rôle in the prediction methodolies. Notice that it may be very simply determined from speed measurements as part of an online prediction algorithm.

The historical average travel time, $\overline{T}(t)$, for a journey starting at time of day, t, is given by

$$\overline{T}(t) = \frac{1}{|D|} \sum_{d \in D} T(d,t) \tag{2}$$

where $|D|$ is the number of days in the set D.

The task of a journey time prediction method is to *estimate* $T(d,t+\delta)$ for time lag $\delta > 0$ *given only* information known at time t on day d. Time t is the *decision time* for estimating a journey beginning after a *lag* of δ at time $t+\delta$.

Two naïve estimates of the journey time, $T(d,t+\delta)$, are

1. $T^*(d,t)$, the *frozen-field* estimator evaluated at the decision time, t, and
2. $\overline{T}(t+\delta)$, the *historical mean* estimator for journeys starting at time (of day) $t+\delta$.

The frozen-field estimator, $T^*(d,t)$, assumes, therefore, that speeds remain held permanently fixed at their time t values throughout the journey. We would expect that this estimator would behave best at small values of δ, where it is able to capture from the real-time measurements known up to time t specific features of the traffic profile on day d. As δ increases these (frozen) features become less relevant compared to the information captured by the long-run historical average estimator, $\overline{T}(t+\delta)$.

2.2 Linear regression method using varying coefficients

Rice & van Zwet observed in US loop detector data a strong linear relationship between the frozen field estimator, $T^*(d,t)$, and the exact observed journey time, $T(d,t+\delta)$, of the form

$$T(d,t+\delta) = \alpha(t,\delta) + \beta(t,\delta) T^*(d,t) + \epsilon \tag{3}$$

where ϵ is a mean zero random variable and the coefficients $\alpha(t,\delta)$ and $\beta(t,\delta)$ vary with both the decision time, t, and the lag before the journey begins, δ. Further details of such varying coefficients models are given by Hastie & Tibshirani [4]. The parameters of such a linear model may be fitted through a weighted least squares procedure which minimizes

$$\sum_{d \in D, s \in T} \left(T(d,s) - \alpha(t,\delta) - \beta(t,\delta) T^*(d,t)\right)^2 K(t+\delta+s) \tag{4}$$

where $K(\cdot)$ is the Gaussian density with mean zero and variance σ^2 given by

$$K(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \,. \tag{5}$$

The purpose of the Gaussian density, $K(\cdot)$, is to produce smoothed estimates of the regression coefficients $\widehat{\alpha}(t,\delta)$ and $\widehat{\beta}(t,\delta)$ as both the decision time, t, and the lag, δ, vary. The degree of smoothing is adjusted by the choice of the variance parameter σ. The methodology then yields a *regression-based* journey time estimator, $\widehat{T}(d, t+\delta)$, given by

$$\widehat{T}(d, t+\delta) = \widehat{\alpha}(t,\delta) + \widehat{\beta}(t,\delta) T^*(d,t) \,. \tag{6}$$

Observe that putting $\alpha(t,\delta) = \alpha'(t,\delta)\overline{T}(t+\delta)$ shows that the estimator, $\widehat{T}(d,t+\delta)$, is, in fact, a particular data-dependent linear combination of the two naïve estimators.

2.3 Nearest neighbour methods

In the simple nearest-neighbour method, the estimator of journey time, $T(d,t+\delta)$, is given by finding the previous day, d', which most closely matches the observed speeds up to time t on day d, according to some well-defined distance measure. Hence, if day d' minimizes the distance to d among all previous days then the nearest neighbour estimator, $T^{NN}(d,t+\delta)$, is given by

$$T^{NN}(d,t+\delta) = T(d',t+\delta) \,. \tag{7}$$

Rice & van Zwet[1] offer several alternatives of the distance, $m(d_1,d_2)$, between two days d_1 and d_2. Two such alternatives considered for evaluation are given as follows

$$m_1(d_1,d_2) = \sqrt{\sum_{t-w\le s\le t} \left[T^*(d_1,s) - T^*(d_2,s)\right]^2} \tag{8}$$

and

$$m_2(d_1,d_2) = \sum_{\ell\in L,\, t-w\le s\le t} |V(d_1,\ell,s) - V(d_2,\ell,s)| \tag{9}$$

where w is a *window size* parameter.

The nearest neighbour method can be readily extended to the k-nearest neighbour (k-NN) method. First, the k closest days, $d_1, d_2, \ldots, d_k$ are found. Then, the predictors derived from each similar day are combined in a weighted-averaging scheme, where the weights are proportional to the distance of each day to the present day, d. The predictor for the k-NN method,

[1]Rice & van Zwet also consider a third class of estimators based on a principal components procedure. We have not considered such estimators here as Rice & van Zwet did not find them to improve over the regression or nearest neighbour estimators.

$T^{kNN}(d, t+\delta)$, is hence given by

$$T^{kNN}(d, t+\delta) = \sum_{i=1}^{k} w_i T(d_i, t+\delta) \tag{10}$$

where $w_i = \frac{m(d,d_i)}{\sum_{j=1}^{k} m(d,d_j)}$ and the distance function is $m(d_1, d_2)$. Note how the simple nearest neighbour method is equivalent to the k-NN method with $k = 1$.

Notice that determining the estimator T^{kNN} involves evaluating a distance for each day according to the distance function as well as ranking those distances to find the k closest days.

3 Numerical results

3.1 The MIDAS dataset

The data considered in this report consists of speed measurements collected per minute from 63 MIDAS loop detectors located on lane 2 (where the slow lane is numbered 1) of the clockwise carriageway between junctions 9 and 14 on the M25 London orbital motorway. The spacing between the loops, d_ℓ, is taken as 500m. The data considered ranged from 05:00 to 20:00 (that is, 900 one minute intervals) on weekdays in 2003. Missing values reduced the original 261 weekdays down to 231 days[2]. The split between days of the week was 39 Mondays, 142 midweek days and 50 Fridays. The resulting data formed a velocity field $V(d, \ell, t)$ with dimensions $231 \times 63 \times 900$.

For comparison, the study by Rice & van Zwet included 34 days and 116 loop detectors along 48 miles of I-10 in Los Angeles.

Figure 1 shows a spatio-temporal plot of the speeds for a single day (Monday, 6 January 2003). During the period 06:30 to 10:00, and for much of the road under consideration, vehicles are travelling at relatively low speeds with a backward-propagating wave pattern in the speed profile. Horizontal stripes can be seen in the plot to roughly coincide with bottlenecks forming in the vicinity of junctions.

3.2 Journey times

From the velocity field a travel time, $T(d, t)$, can be constructed for the journey from loop 1 to loop 63 which starts at time t on day d. Figure 2 shows in the top panel how the journey times vary during the day for each of the individual 39 Mondays. Journey times are naturally seen to increase during the morning slowdown period. (Several exceptions occur on Bank Holiday

[2]Missing values within the MIDAS speed data that formed significant blocks over time and loops caused that day to be rejected. More commonly, missing values occured throughout parts of the day at one or more non-adjacent sites. Less frequently, many sites produced missing values for just a single minute. In both of these cases, the missing values were imputed by straightforward linear interpolation.

Mondays.) During the middle portion of the day and again between 17:00 and 19:00 there are significant numbers of days when journey times have increased. However, this feature is much less pronounced than it is in the morning. In contrast, the dataset considered by Rice & van Zwet has most congestion in the period from 15:00 onwards.

The lower panel of Figure 2 shows a "box-and-whiskers" plot of the journey times. The central bar shows the median journey time (over the 39 days) and the length of the box shows the interquartile range (that is, from the 25% to the 75% percentiles). The whiskers extend to the furthest data point that is no more than 1.5 times the interquartile range from the box. Any data points outside of the whiskers are plotted individually. In addition, the orange filled dots are the mean journey times. The plot makes clear that not only are the median journey times longer between 06:30 and 10:00 but that the distribution of journey times is much more spread out within this period.

Figures 3 and 4 show journey times on midweek days and Fridays, respectively. The results for Fridays show considerably longer journey times in the afternoons compared with the Mondays. Journey times for midweek days show a wide variation in journey times in both the mornings and the afternoons but the effect is reduced compared with Fridays.

Figures 2–4 illustrate the strong day-of-week effect on journey times and we have used these three categories of weekdays (namely, Mondays, midweek days and Fridays) to separately estimate journey times.

The key linear relationship identified by Rice & van Zwet that underlies the prediction methodology is between $T^*(d,t)$ and $T(d,t+\delta)$. Figure 5 shows scatterplots of these two quantities where the decision time, t, is 08:00 and the lag δ ranges from 0 to 120 minutes and the data is confined to just the 39 Mondays. Each plot also shows the historical mean estimator as a horizontal line. Notice how the slope of the regression line diminishes as the lag increases.

Equation (4) was used to fit the regression coefficients $\alpha(t,\delta)$ and $\beta(t,\delta)$ by a standard weighted least squares procedure. The regression-based journey time estimator $\widehat{T}(d,t)$ was then obtained from the fitted coefficients through equation (6).

Figure 6 shows how the fitted parameters $\alpha(t,\delta)$ and $\beta(t,\delta)$ vary with t and δ. The smoothness of the surfaces is controlled by the parameter σ which here was taken as $\sigma = 10$ minutes. (We discuss at length the choice of such parameters later in Sections 3.3 and 3.4.) The parameter β is seen to increase steeply with the lag during the early rush hour period with simultaneous decreases in α. This is explained by noting that at the start of the rush hour period journey times increase rapidly and, as a result, the frozen-field estimate needs to be multiplied by a larger factor β to better predict future journey times. Figure 7 shows how the linear relationship between the frozen-field estimate and the journey time becomes much steeper, making the slope β large and correspondingly pushing down the intercept parameter α. Accordingly, we would expect an improved estimator if this small number of outlier days was removed and the coefficients fitted to the remaining data. Alternatively, a robust form of regression could be used in place of the least-squares approach which is less sensitive to outliers.

Figure 8 repeats the central scatterplot from Figure 5 where the lag is $\delta = 60$ minutes. The

central sloping line gives the regression estimator, $\widehat{T}$, for the journey time as a function of the frozen field estimator T^*. An important consequence that would follow from the adoption of Gaussian errors in the statistical model for $\widehat{T}$ in (3) is that the many powerful techniques and tools of Gaussian models can then be applied. In particular, the same statistical model may also be used to construct a *prediction interval* (shown in Figure 8 by the outer pair of sloping lines). The prediction interval illustrated here gives a region that we expect, given the statistical model, to contain the exact journey time with a probability of 90%. The level of 90% is for illustration only. It could either be higher or lower corresponding to intervals that are wider or narrower, respectively.

It may be worth concluding this section by describing how the regression estimator would be implemented. Using historical data, such as that shown in Figure 2, the regression model is fitted and the sloping lines on Figure 8 are computed. This part of the calculation is done offline and the results are saved for use by the online part of the algorithm. At the decision time, t, the frozen field estimator T^* is obtained from the current speed measurements (in our example journey this involves a simple calculation (given by equation (1)) using the speed values recorded by the 63 MIDAS loop detectors). The regression estimator $\widehat{T}$ and the prediction interval are then looked up from the saved results of the offline calculation. For the example shown in Figure 8, if the online calculation of T^* yields a value of 30.00 minutes then the regression estimator is $\widehat{T} = 22.31$ minutes and the 90% prediction interval is (15.39,29.24). If the frozen field estimator was instead a value of 60.00 then the regression estimator would be $\widehat{T} = 34.51$ minutes and the 90% prediction interval would be (27.56,41.45). The historical mean estimator, $\overline{T}$, is computed from historical measurements alone and in both these cases, independent of online measurements, it is 28.08 minutes.

3.3 Comparison of methodologies

Figure 9 shows how the root-mean-square prediction errors on Mondays for our four estimators varies as t varies throughout the period between 05:00 and 20:00 and with the lags, δ, increasing from 0 to 120 minutes. The historical mean estimator is not affected by the choice of lag, δ, except that the curves shown shift leftwards by the amount δ. The frozen-field estimator has larger root-mean-square prediction error as the lag, δ, increases and the relative importance of recent information recedes. The regression-based estimator has the lowest root-mean-square prediction error. During the period 6:30 to 10:00 it has more than halved the error compared to the historical mean. Later in the day, when journey times are far less variable there is little benefit to be obtained from the regression approach compared to simply using the historical mean. As the lag, δ, is allowed to increase the error in the regression-based estimator, $\widehat{T}$, approaches that of the historical mean. The frozen-field estimator, T^*, can have a large prediction error for even moderate values around 30 minutes of the lag. Figure 9 also includes the nearest neighbour estimator T^{kNN} calculated with $k = 4$, a window size parameter of $w = 20$ minutes and the $m_1(\cdot)$ distance function. The performance of the T^{kNN} estimator is quite similar to the regression estimator.

Figures 10 and 11 show the prediction errors for the cases of midweek days and Friday, respectively. A similiar comparison applies in these two categories. However, the prediction error with the historical mean estimator is rather greater in the case of Friday afternoons than occurs on the Mondays. Therefore, there is considerable scope for using real-time information to reduce the prediction error of journey times as can be seen with both the regression and nearest neighbour estimators.

Figures 9–11 taken together show that when the prediction error in the historical mean is high it is possible for the regression and nearest neighbour methods to dramatically reduce the prediction error, at least for short to medium lags. For longer lags, over 2 hours (say), all estimators will finally approach the performance of the historical mean.

It is quite surprising that despite investigating a wide choice of parameters (k and w for the nearest neighbour estimator and σ for the regression estimator) we were unable to observe any significant improvement of the nearest neighbour procedure over the regression procedure. The regression procedure has rather minimal online requirements as discussed above compared to the nearest neighbour procedure which must compute an online search for the k closest days.

3.4 Validation of parameter choices

We now discuss the approach followed to select the parameter values used above. For the case of the regression estimator we must select the smoothing parameter σ. Figure 12 shows how the prediction errors for the Mondays varied as σ was allowed to vary within the range from 5 to 100 minutes. Variation of σ within the range from 5 to 20 had little effect on the prediction errors. Only when σ was allowed to increase further to 50 and 100 was there any noticeable deterioration in the prediction error. Hence, our selection of $\sigma = 10$ minutes used earlier.

For the nearest neighbour method there are rather more parameters to select. There is the choice of k, the number of closest neighbours to consider, and the window size parameter w. In addition, there is the choice of distance function, $m_1(\cdot)$ or $m_2(\cdot)$ to use.

Figures 13–15 concern the effects of k and w when the $m_1(\cdot)$ distance function is used. Conversely, Figures 16–18 use the $m_2(\cdot)$ distance function. For both distance functions the effects of k and w are similar. As k increases from 1 to about 4 there is a small improvement in the prediction error but beyond 4 as k increases further to 25 the prediction error grows slighly again. Hence, our use earlier of $k = 4$. In the case of the window size parameter w the optimal choice appears to be around 20.

Finally, the choice of distance function itself appears to have little effect and we have chosen to work with $m_1(\cdot)$ which involves the frozen field quantities directly rather than the speeds.

4 Conclusions

In this report we describe our findings from a project using MIDAS loop detector data for journey time prediction. We have found that the simple-to-implement regression-based method

of Rice & van Zwet [5] works well in our example scenario of UK data taken from the M25 London orbital motorway in 2003. Phase one of the project involved a short study of the data formats used by MIDAS to record traffic count data and described a revised data format including explicit indexing. This revised format, based on the familiar ZIP file archiving tool allowed efficient random access to the data necessary for high throughput applications.

The project looked at the variability of journey times across days in three day categories: Mondays, midweek days and Fridays. The regression-based estimator together with a k-nearest neighbour estimator were studied and the results compared in terms of the root-mean-square prediction error. It was found that where the variability was greatest (typically during the rush hours periods or periods of flow breakdowns) the regression and nearest neighbour estimators reduced the prediction error substantially compared with a naïve estimator constructed from the historical mean journey time. Only as the lag between the decision time and the journey start time increased to beyond around 2 hours did the potential to improve upon the historical mean estimator diminish. Thus, there is considerable scope for prediction methods combined with access to real-time data to improve the accuracy in journey time estimates. In so doing, they reduce the generalised cost of travel. The regression-based prediction estimator has a particularly low computational overhead, in contrast to the nearest neighbour estimator, which makes it entirely suitable for an online implementation.

Finally, the project demonstrates both the value of preserving historical archives of transport related datasets as well as provision of access to real-time measurements.

References

[1] R.J. Gibbens and Y. Saatci. Road traffic analysis using MIDAS data: Phase one interim report: data organization. Computer Laboratory, University of Cambridge. DfT Horizons project H05-217, February 2006.

[2] R.J. Gibbens and Y. Saatci. Road traffic analysis using MIDAS data: Phase two interim report: journey time prediction. Computer Laboratory, University of Cambridge. DfT Horizons project H05-217, June 2006.

[3] R.J. Gibbens and W. Werft. Data gold mining. *Significance*, 2(3):102–105, September 2005.

[4] T. Hastie and R. Tibshirani. Varying coefficients model. *J. R. Stat. Soc. B.*, 55(4):757–796, 1993.

[5] John Rice and Erik van Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, September 2004.

[6] W. Werft. Travel time prediction in road networks. MPhil in Statistical Science, Statistical Laboratory, University of Cambridge, 2005.

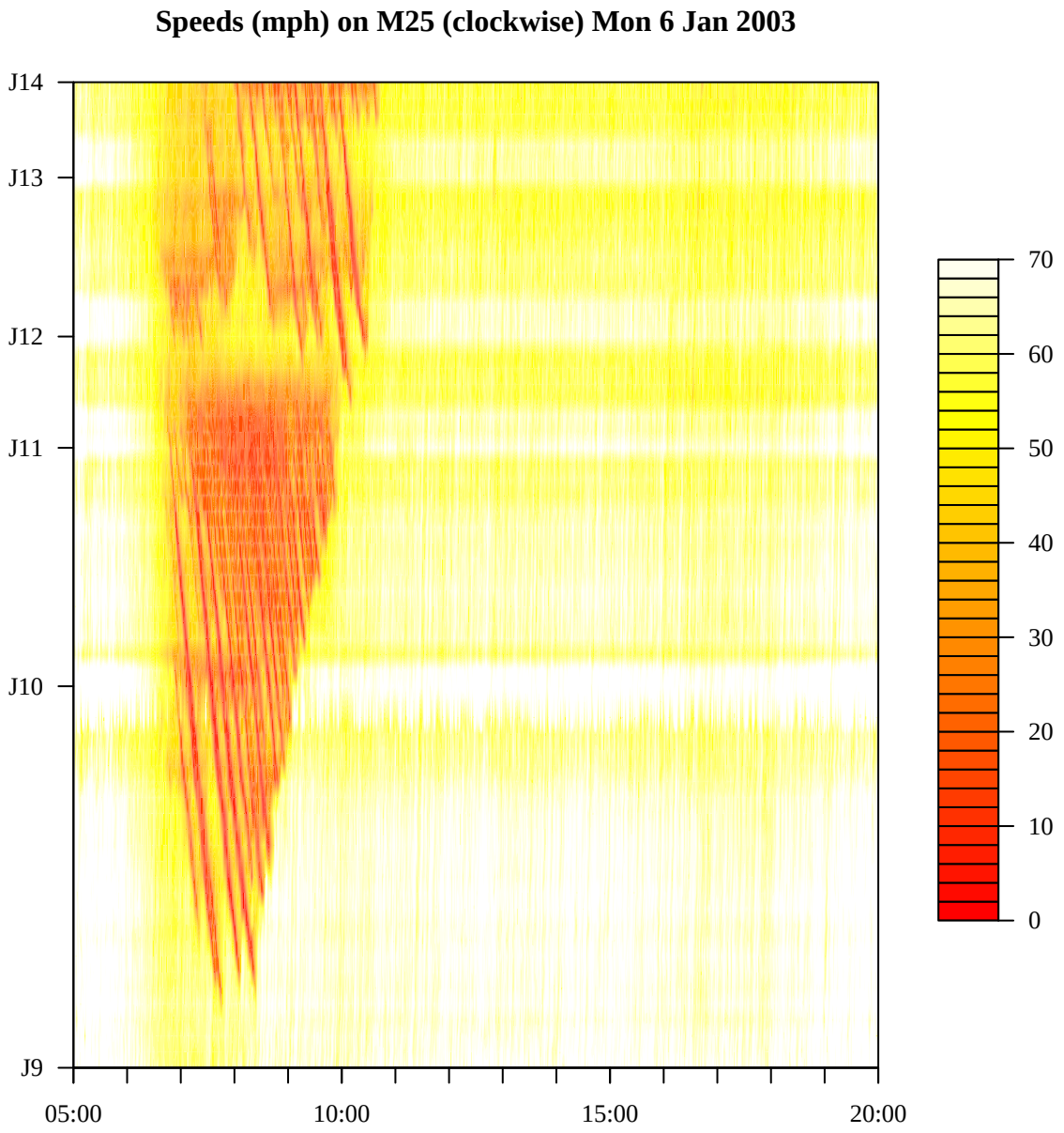


Figure 1: A spatio-temporal plot of the speeds (measured in mph) on lane 2 of the clockwise carriageway of the M25 between junctions 9 and 14 on Monday, 6 January 2003. There is a region of severe congestion in the morning rush hour where speeds are much reduced and have a backward-propagating wave-like profile. Bottlenecks roughly coincide with junctions as shown by the horizontal stripes.

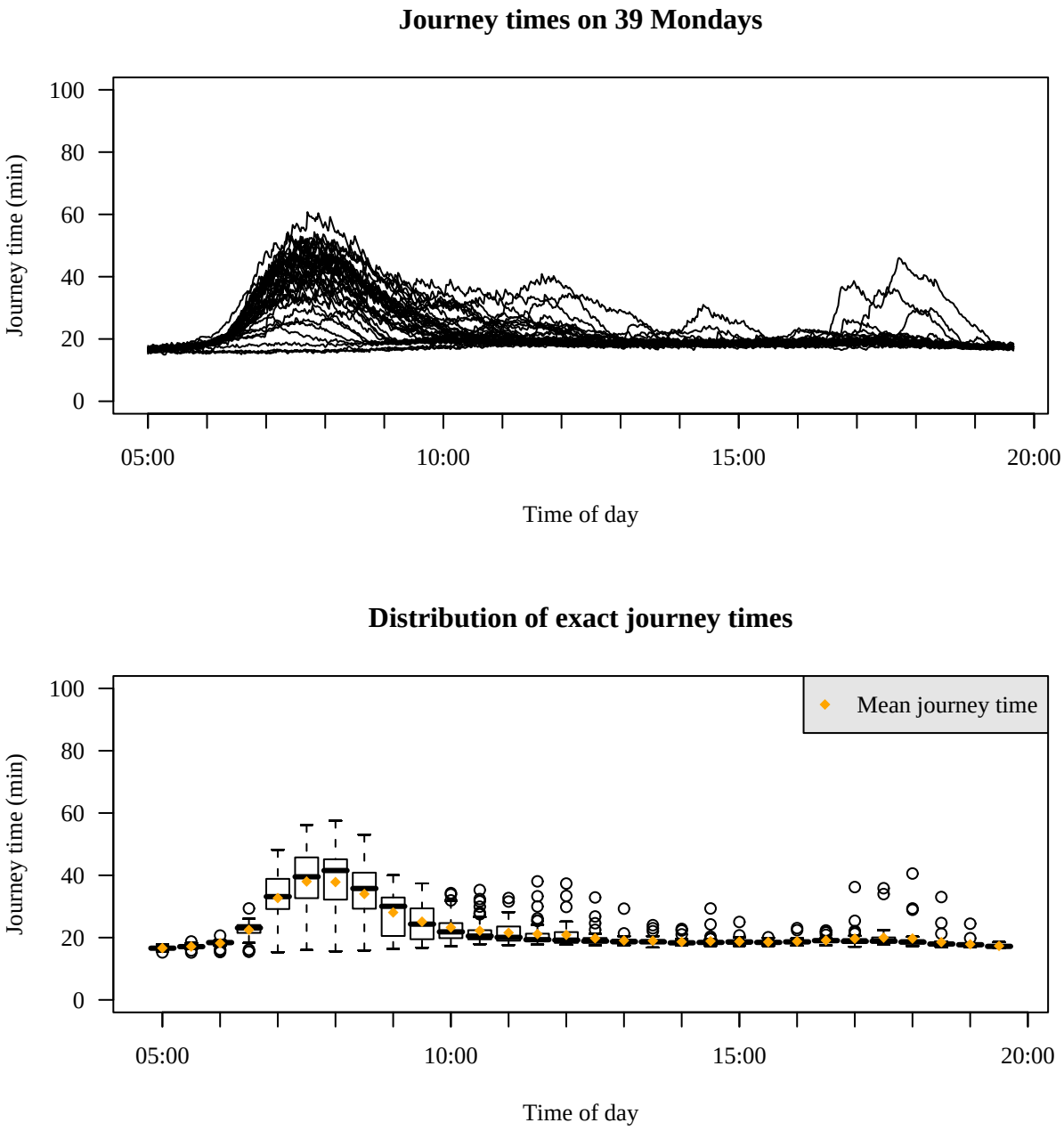


Figure 2: The top panel shows journey times on 39 Mondays during 2003 starting at times ranging from 05:00 to 20:00. The lower panel shows the distribution of journey times by means of box-and-whiskers plots. Journey times are not just longer during the morning rush hour period but also more spread out.

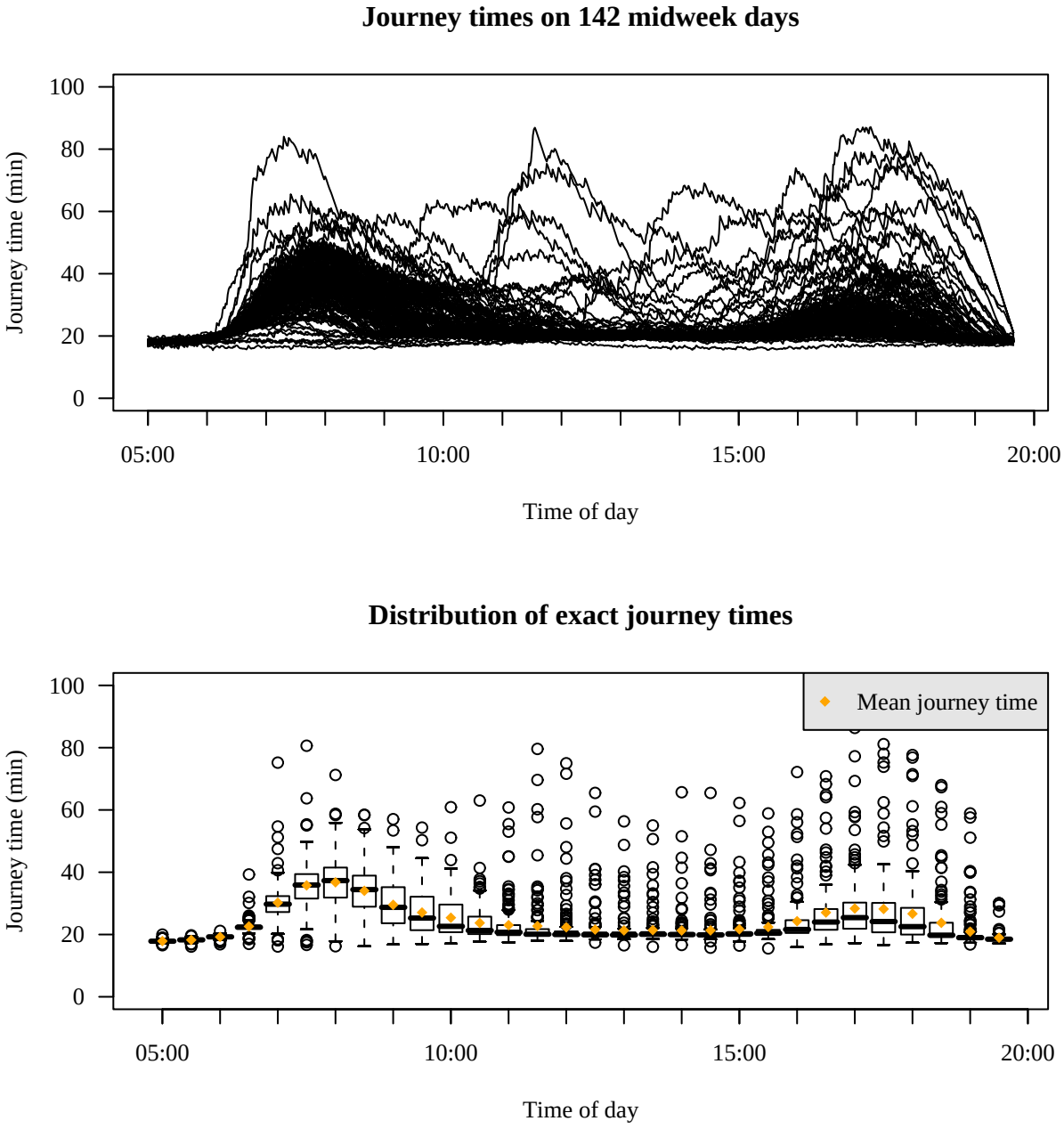
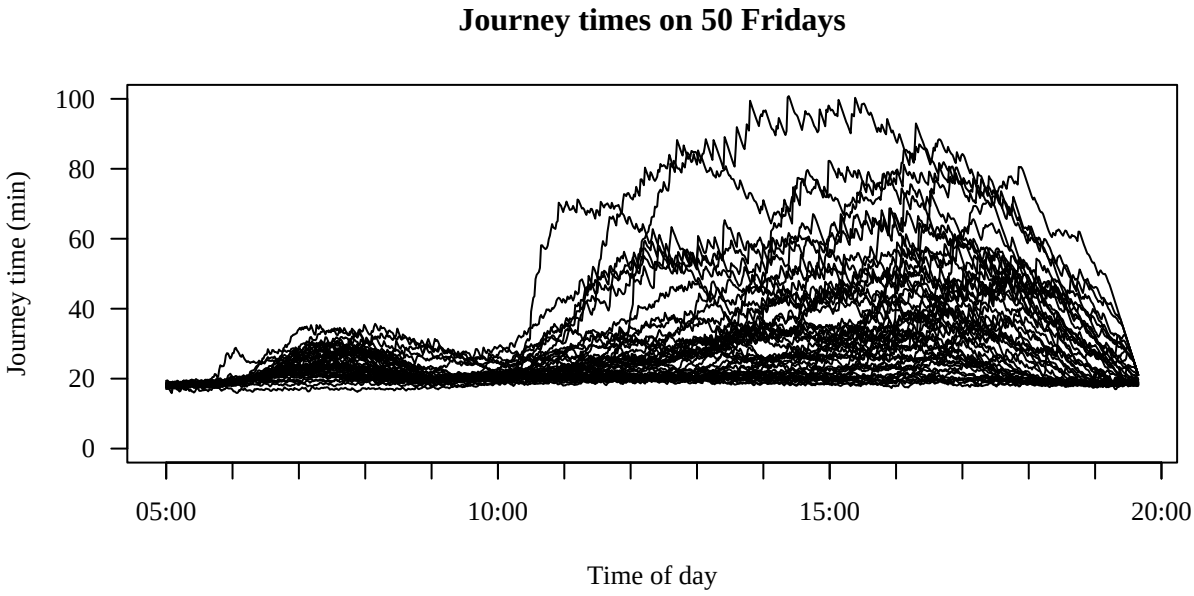


Figure 3: The top panel shows journey times on 142 midweek days (Tuesday, Wednesday and Thursday) during 2003 starting at times ranging from 05:00 to 20:00. The lower panel shows the distribution of journey times by means of box-and-whiskers plots. Median journey times rise during the morning and evening rush hours and there are many outlier days with longer journey times.



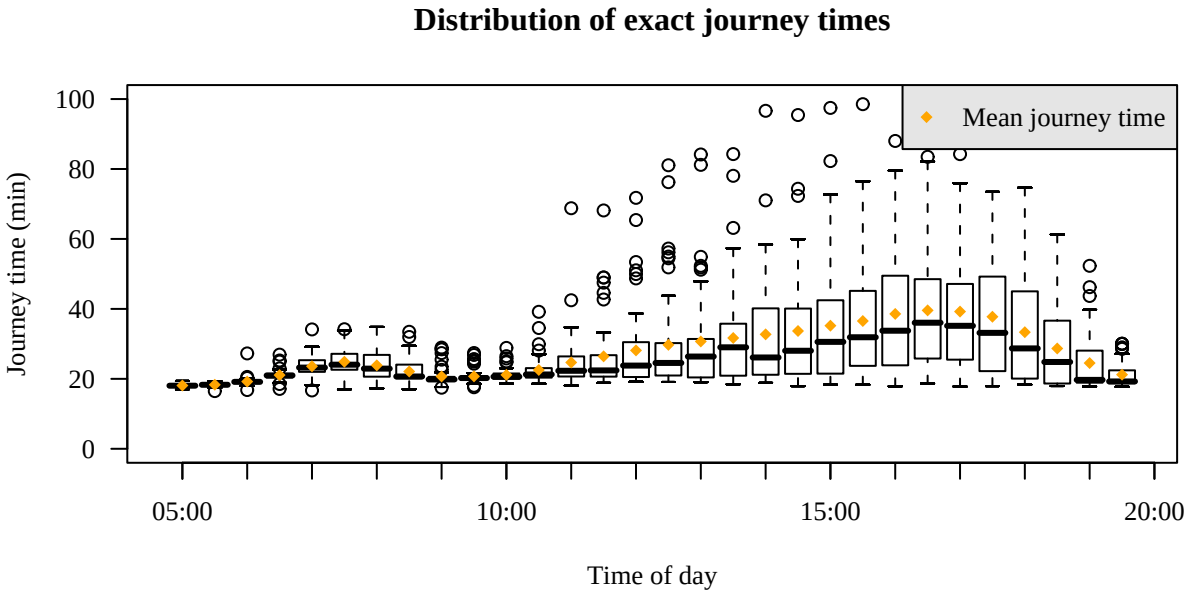


Figure 4: The top panel shows journey times on 50 Fridays during 2003 starting at times ranging from 05:00 to 20:00. The lower panel shows the distribution of journey times by means of box-and-whiskers plots. Median journey times rise significantly from mid-day onwards along with a very wide variation in journey times.

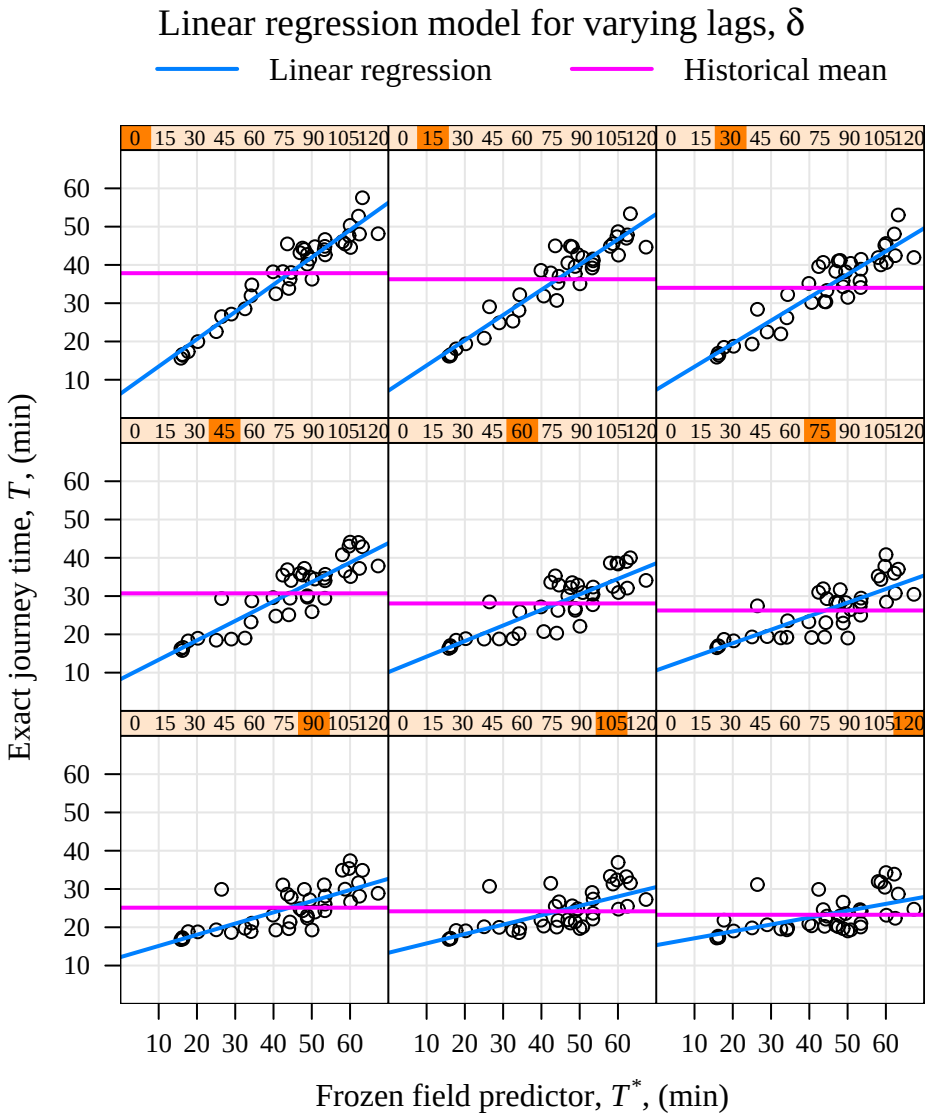


Figure 5: The figure illustrates the linear relationship between the frozen-field estimator $T^*(d,t)$ and the journey time $T(d,t+\delta)$. Here the decision time, t, is fixed at 8:00 on Mondays and the lag, δ, increases from 0 to 120 minutes. Both the historical mean and least-squares regression are shown.

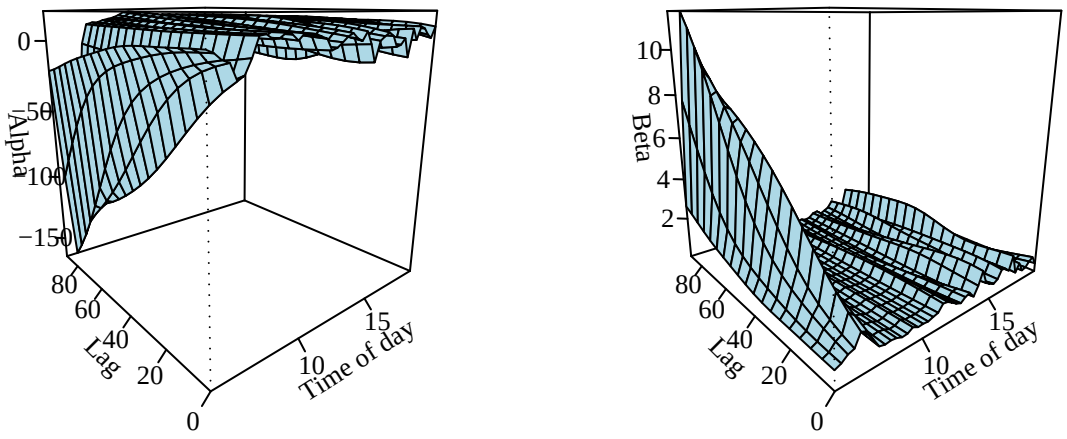


Figure 6: The panel on the left shows the values of $\alpha(t, \delta)$ as the start time, t, varies throughout the period 05:00 to 20:00 and as the lag, δ, increases from 0 to 90 minutes. The panel on the right shows the variation of $\beta(t, \delta)$.

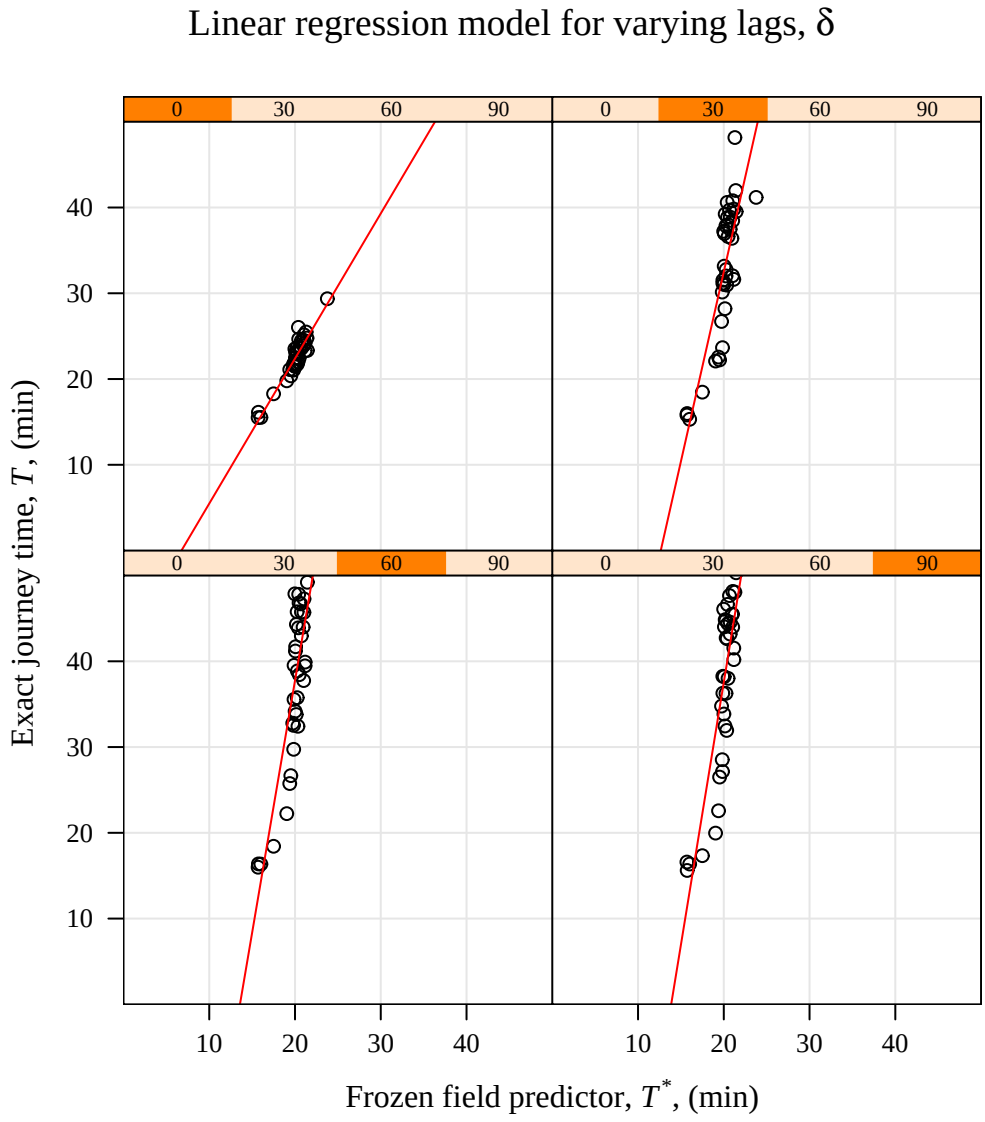


Figure 7: This figure shows how the regression-based method reflects the rapidly increasing journey times during rush hour periods by moving to a more vertical linear relationship.

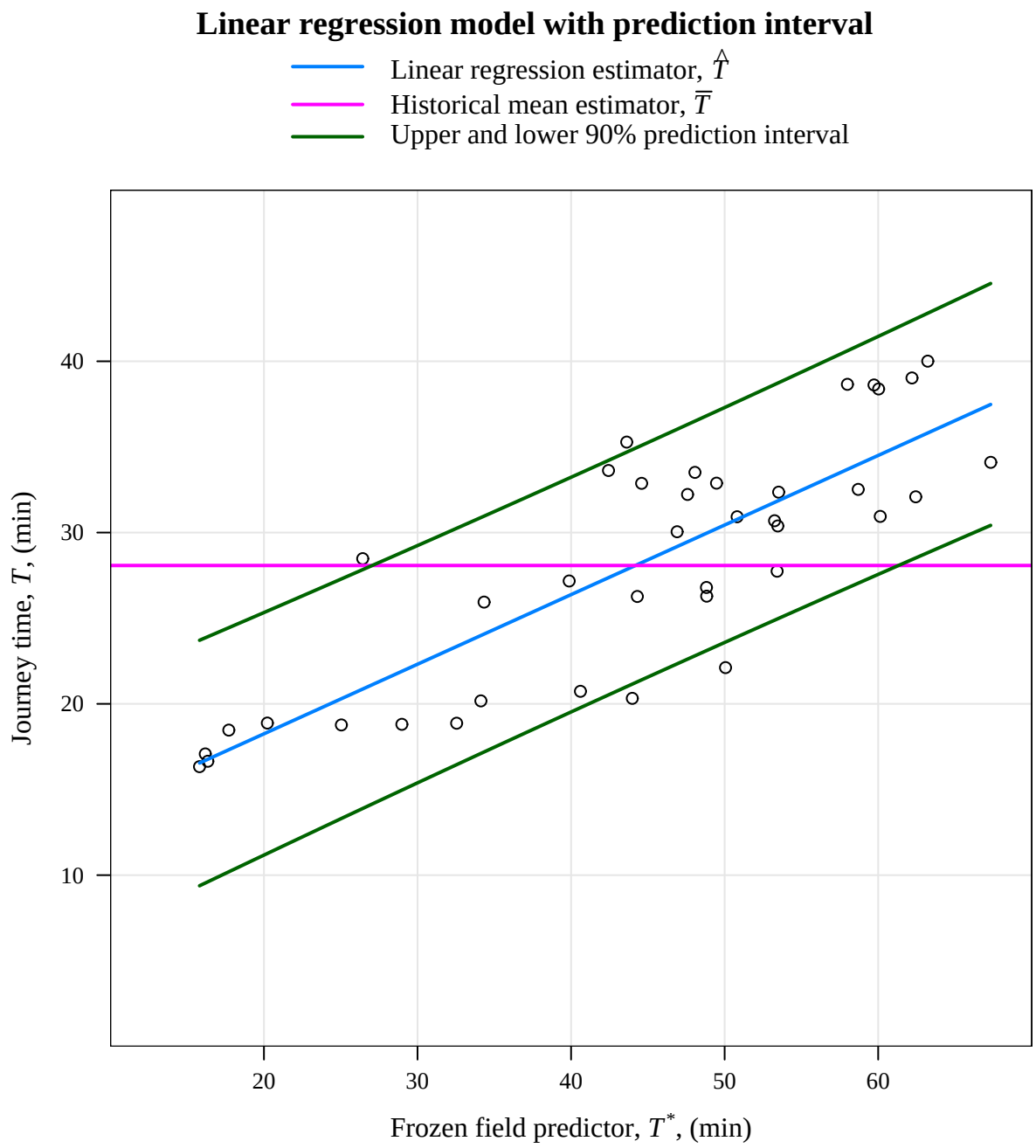


Figure 8: This figure shows properties of a linear model with Gaussian errors. The central sloping line is the regression estimator, $\widehat{T}$ as a function of the frozen-field estimator T^* using the data for Mondays only. The outer pair of sloping lines are a 90% prediction interval for the journey time given a value for the frozen-field estimator. The horizontal line gives the historical mean journey time.

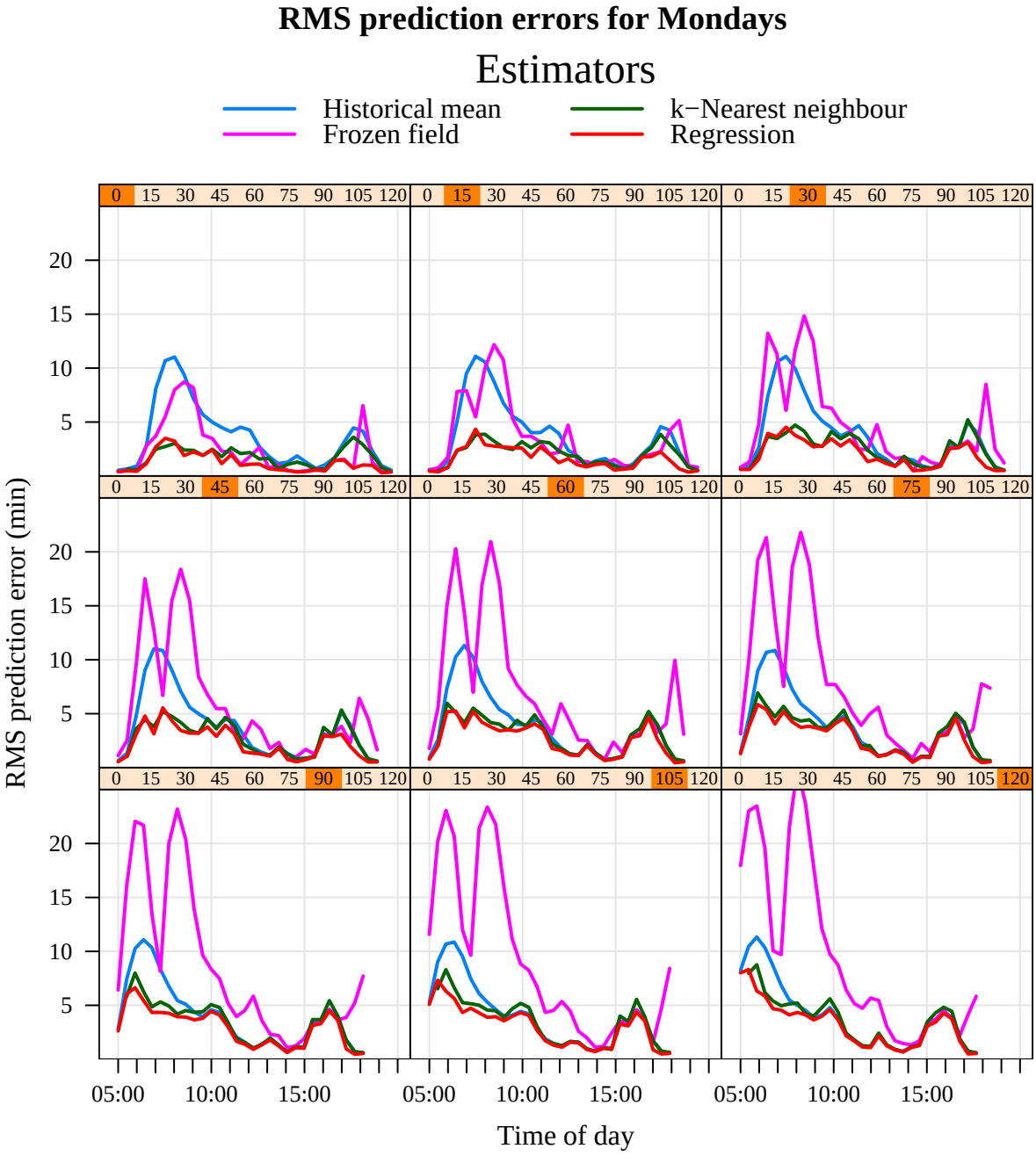


Figure 9: The figure shows the root-mean-square prediction errors for the four estimators with data from Mondays only over the range of start times and as the lag, δ varies from 0 to 120 minutes. The regression-based estimator has improved over the historical and frozen-field estimators. The nearest neighbour estimator appears to compare well to the regression estimator. The benefits in terms of reduced prediction error diminish when the lag becomes large or when there is little inherent variability in the journey times.

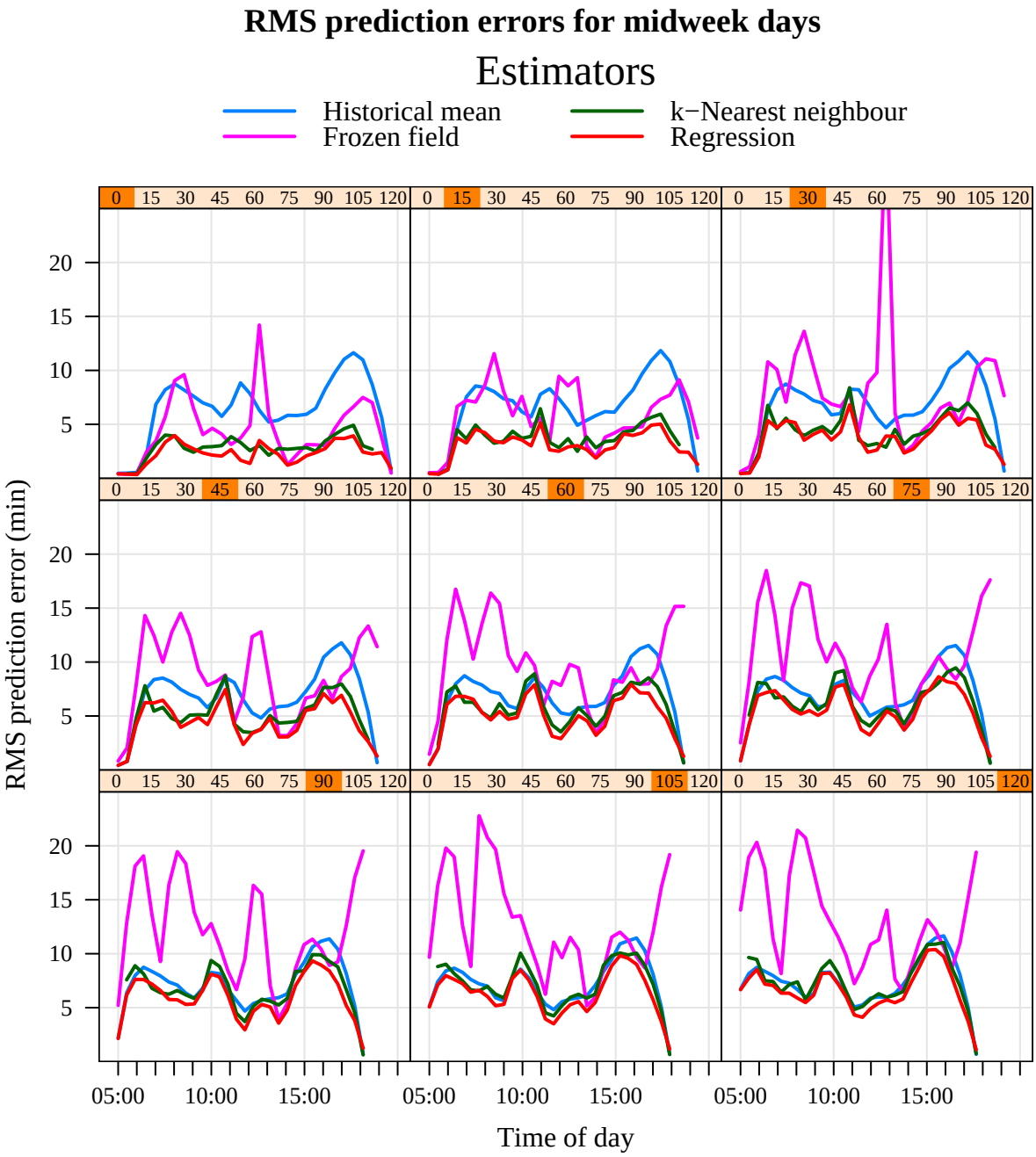


Figure 10: The figure shows the root-mean-square prediction errors for the four estimators with data from midweek days only over the range of start times and as the lag, δ varies from 0 to 120 minutes.

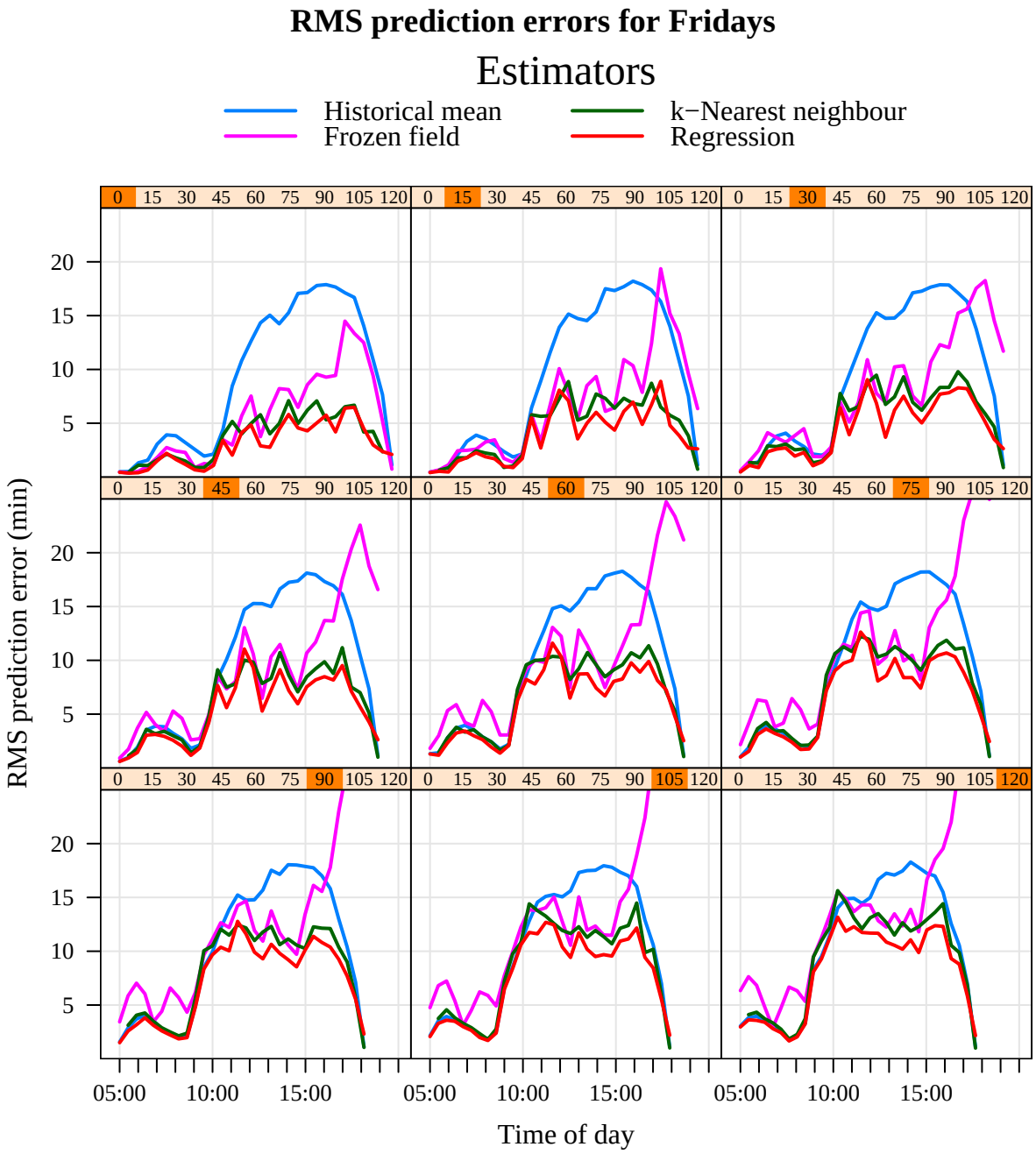


Figure 11: The figure shows the root-mean-square prediction errors for the four estimators with data from Fridays only over the range of start times and as the lag, δ varies from 0 to 120 minutes.

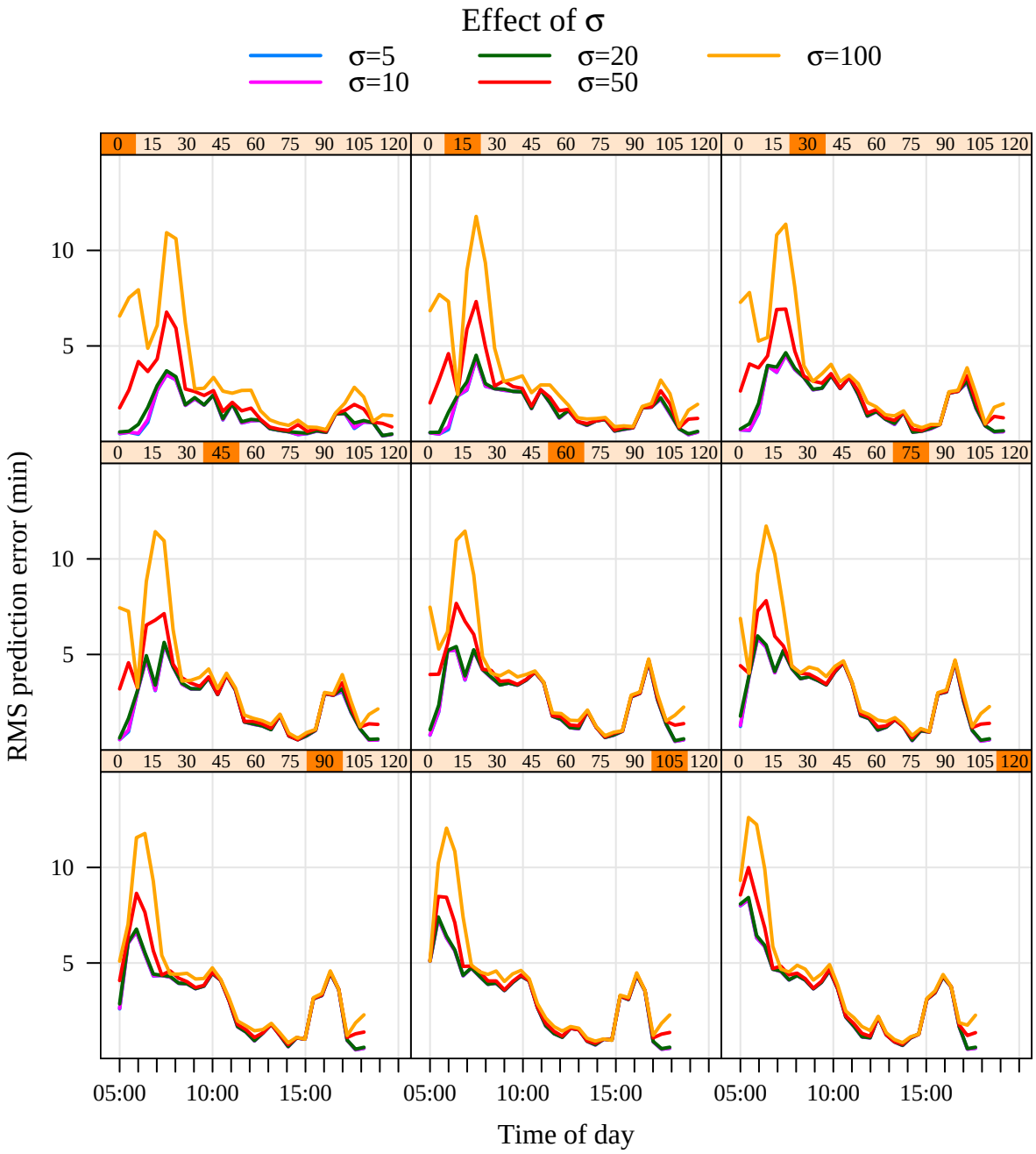


Figure 12: This figure shows for the Monday data the effect of the smoothing parameter σ on the prediction errors of the regression estimator. A choice of $\sigma = 10$ minutes was selected as optimal.

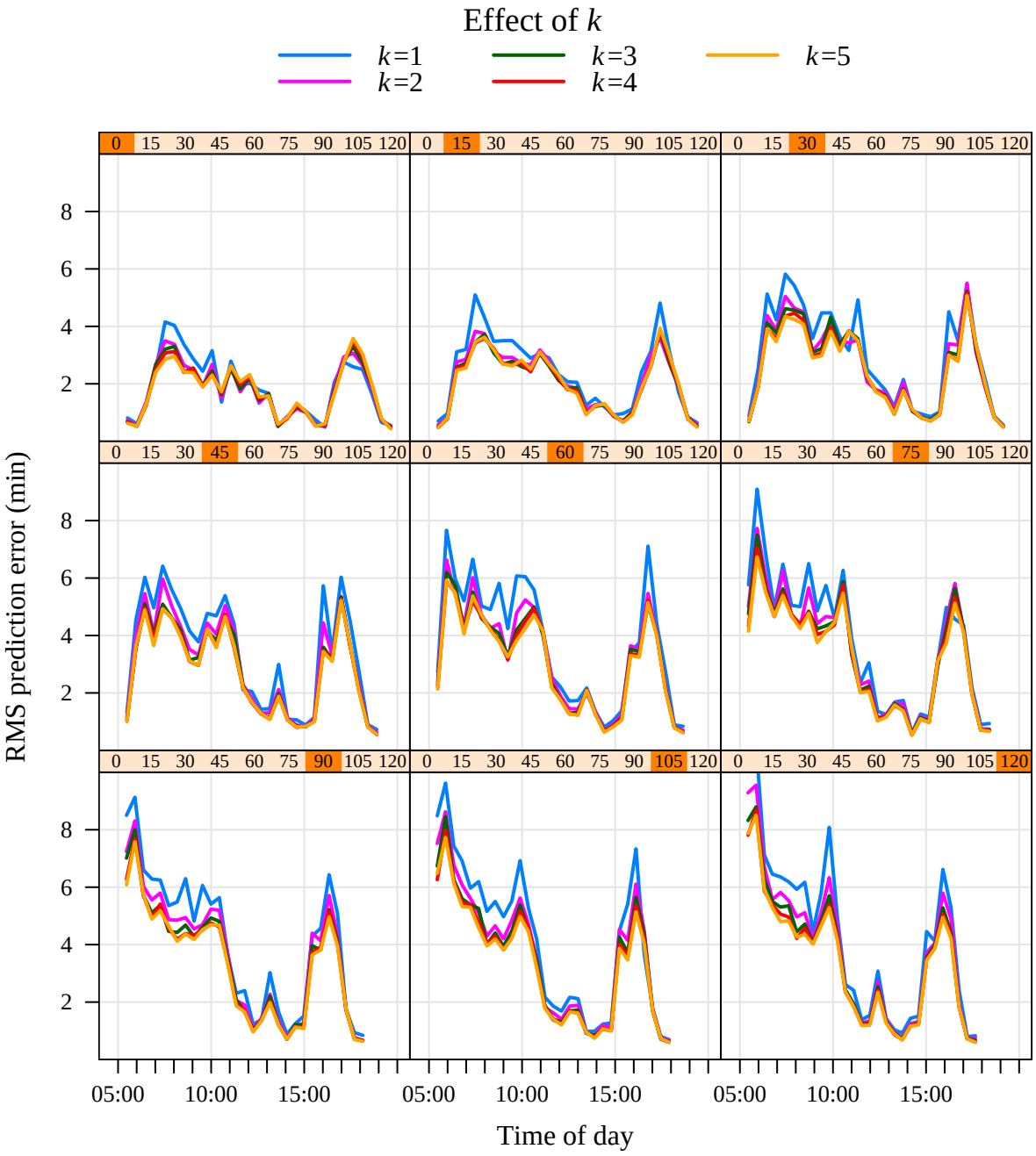


Figure 13: This figure uses the $m_1(\cdot)$ distance function and looks at the effect of the choice of k, the number of neighbours in the nearest neighbour methods. As k increases from 1 to 4 there is a small improvement in the prediction errors. The window size parameter was held fixed at 20.

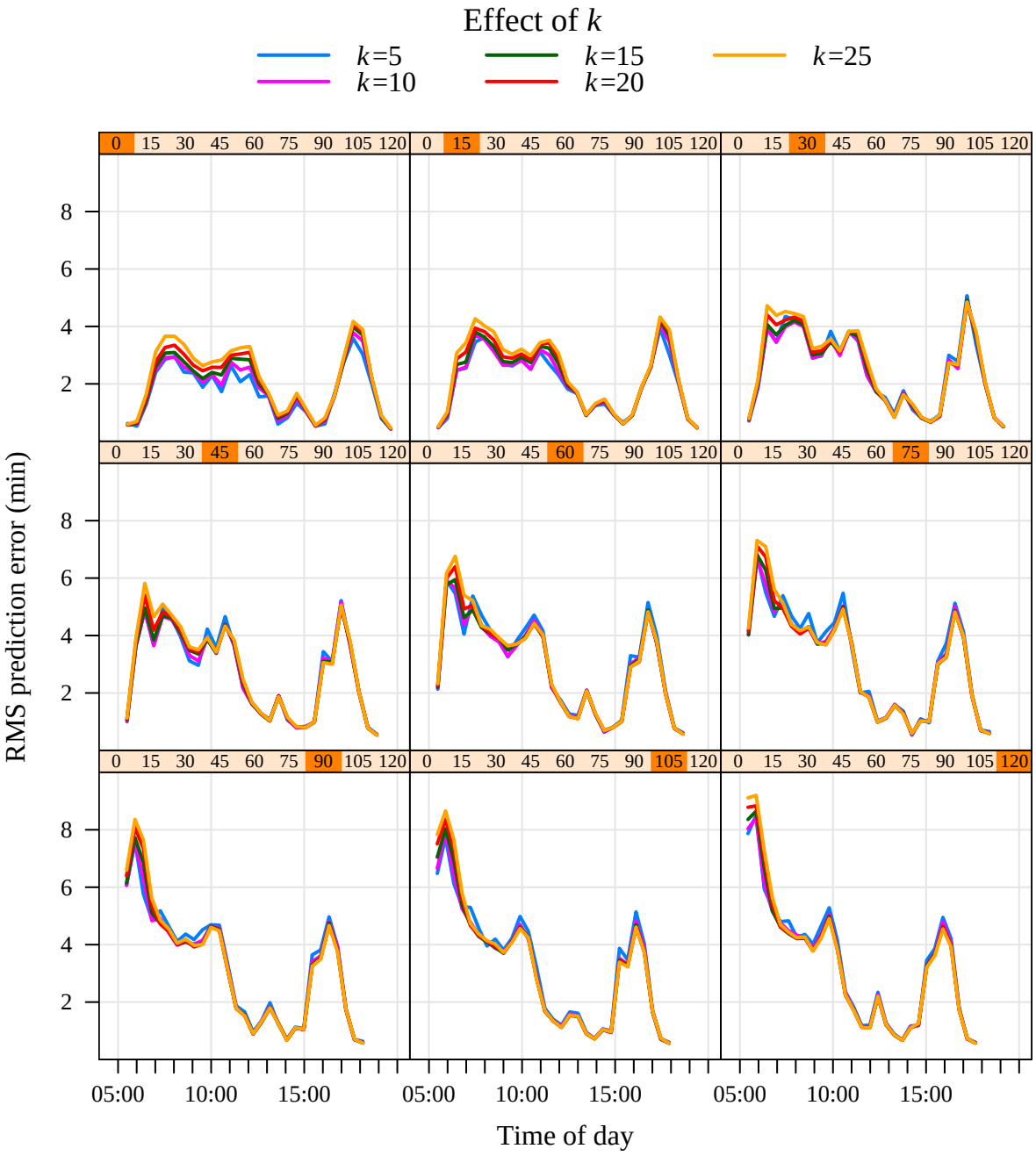


Figure 14: This figure uses the $m_1(\cdot)$ distance function and looks at the effect of the choice of k, the number of neighbours in the nearest neighbour methods. As k increases from 4 to 25 there is a small increase in the prediction errors. The window size parameter was held fixed at 20.

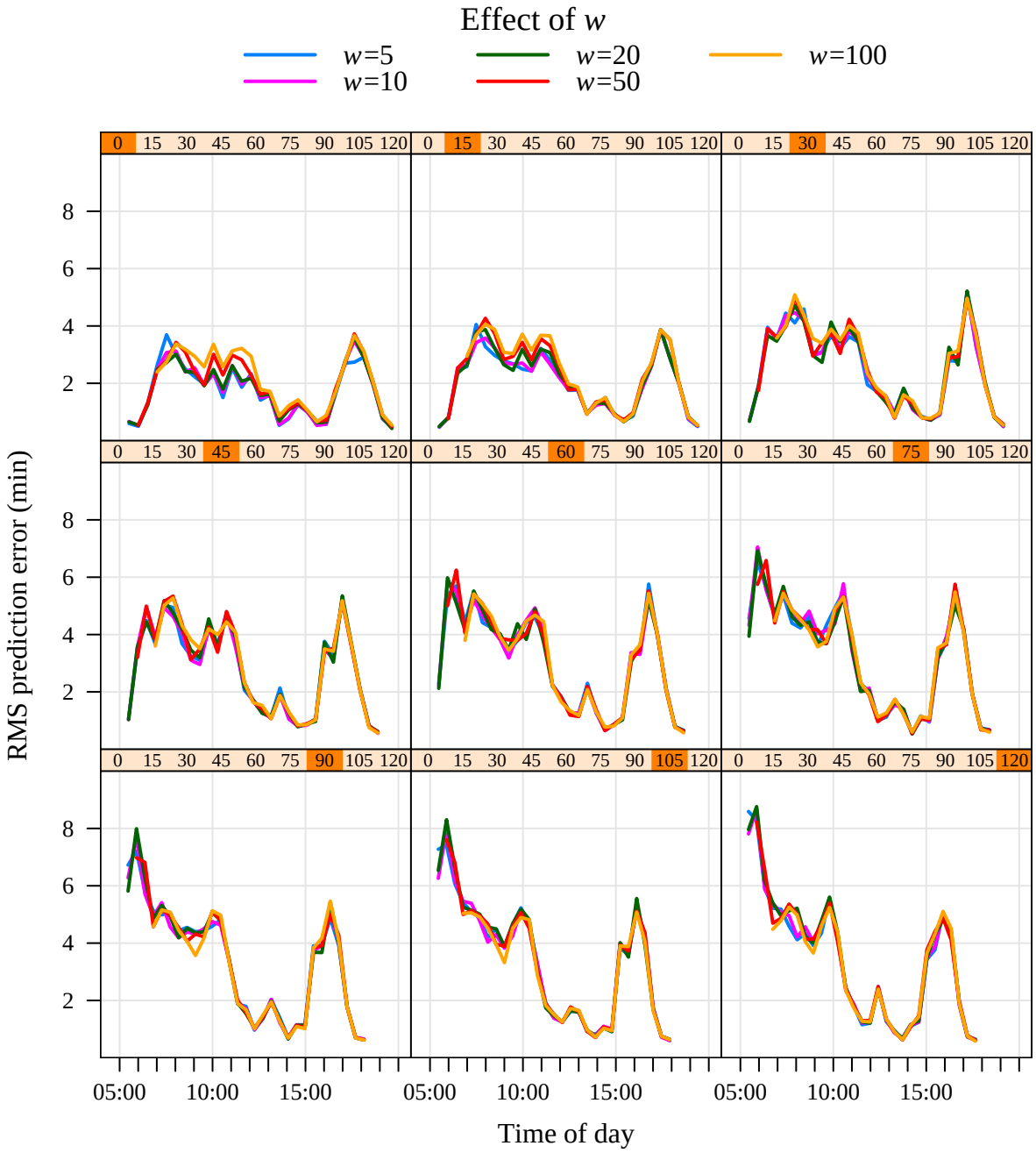


Figure 15: This figure uses the $m_1(\cdot)$ distance function and looks at the effect of the window size parameter w. The prediction error is minimal when w is around 20. The value of k was held fixed at 4.

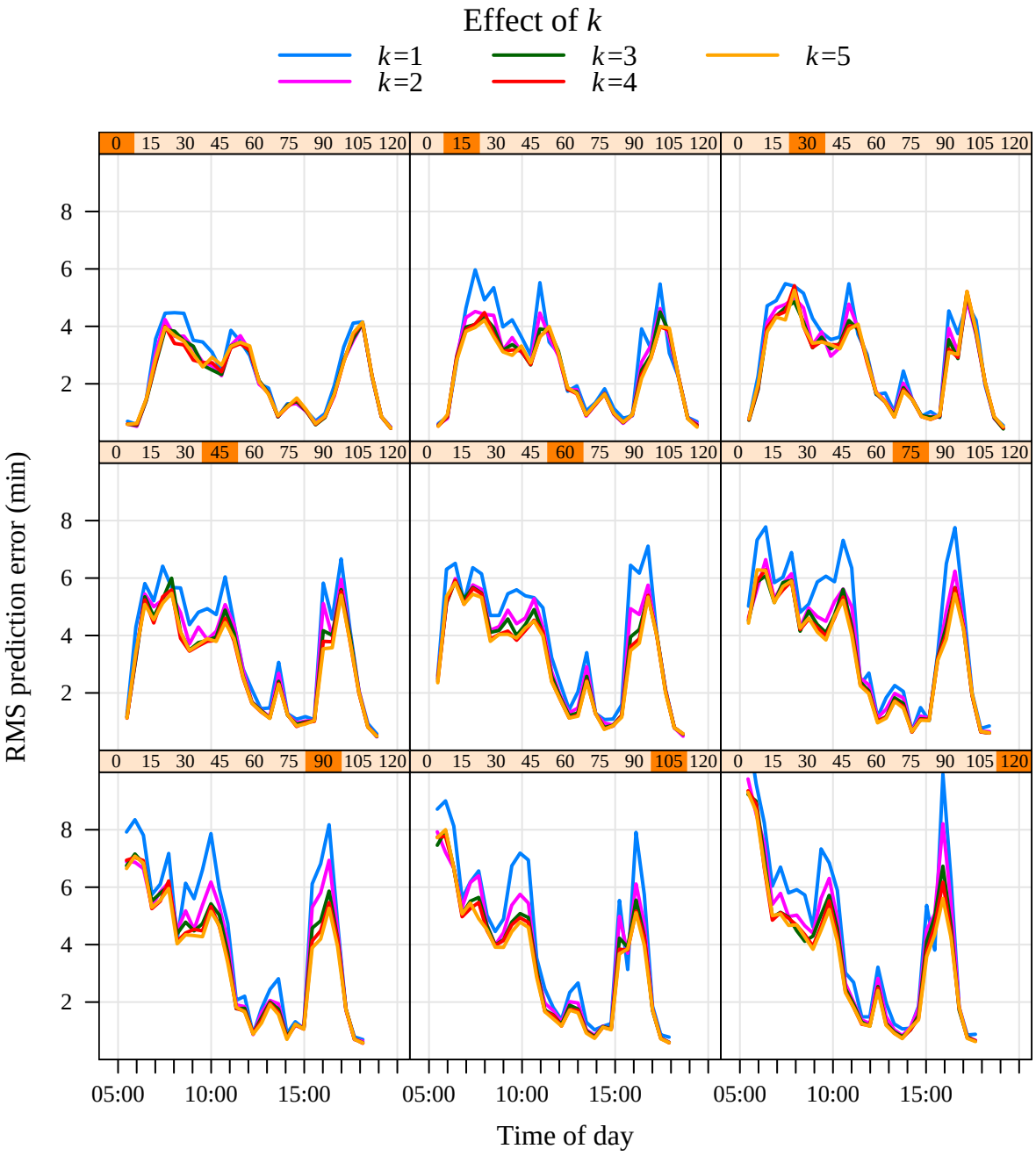


Figure 16: This figure uses the $m_2(\cdot)$ distance function and looks at the effect of the choice of k, the number of neighbours in the nearest neighbour methods. As k increases from 1 to 4 there is a small improvement in the prediction errors. The window size parameter was held fixed at 20.

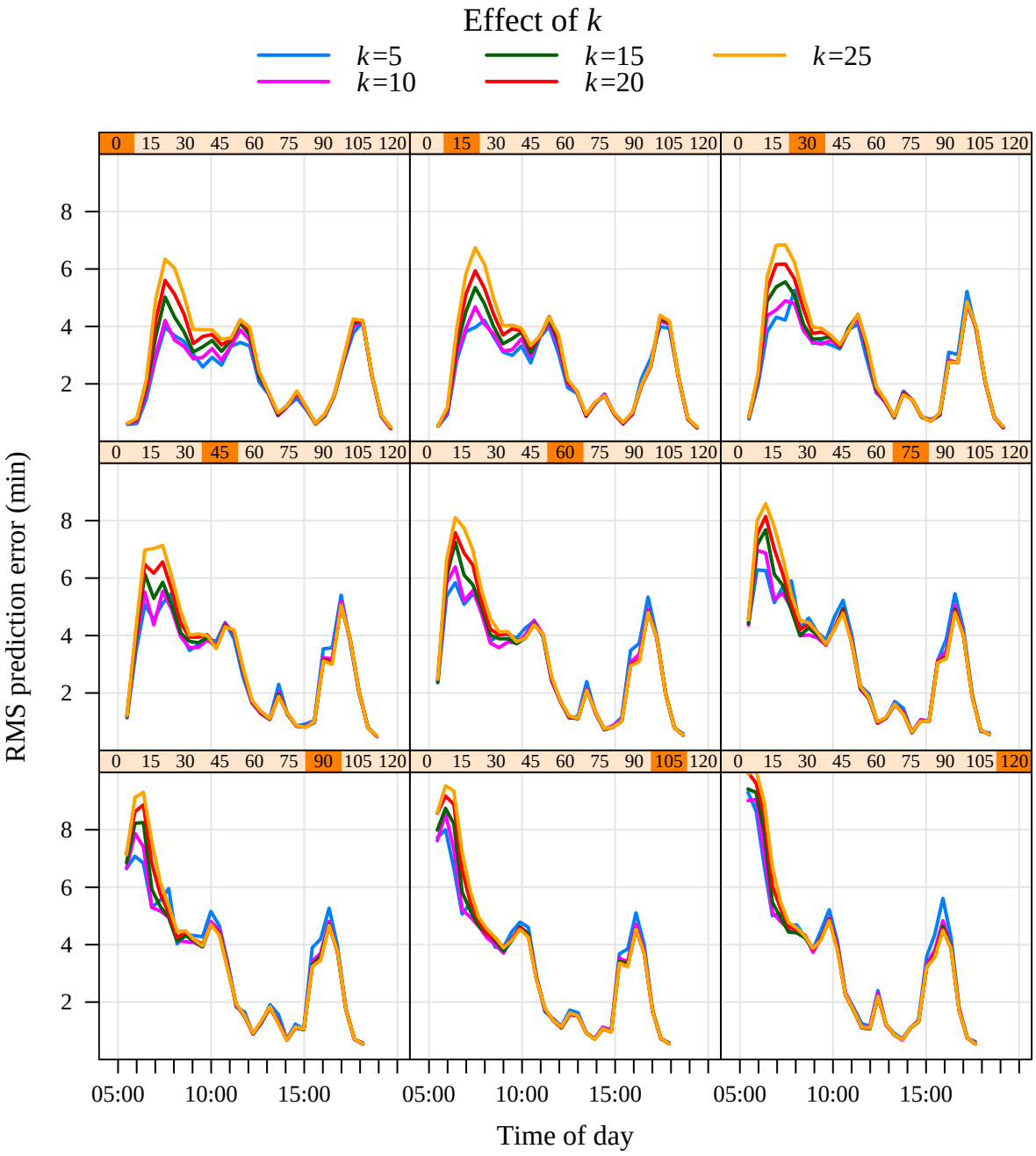


Figure 17: This figure uses the $m_2(\cdot)$ distance function and looks at the effect of the choice of k, the number of neighbours in the nearest neighbour methods. As k increases from 4 to 25 there is a small increase in the prediction errors. The window size parameter was held fixed at 20.

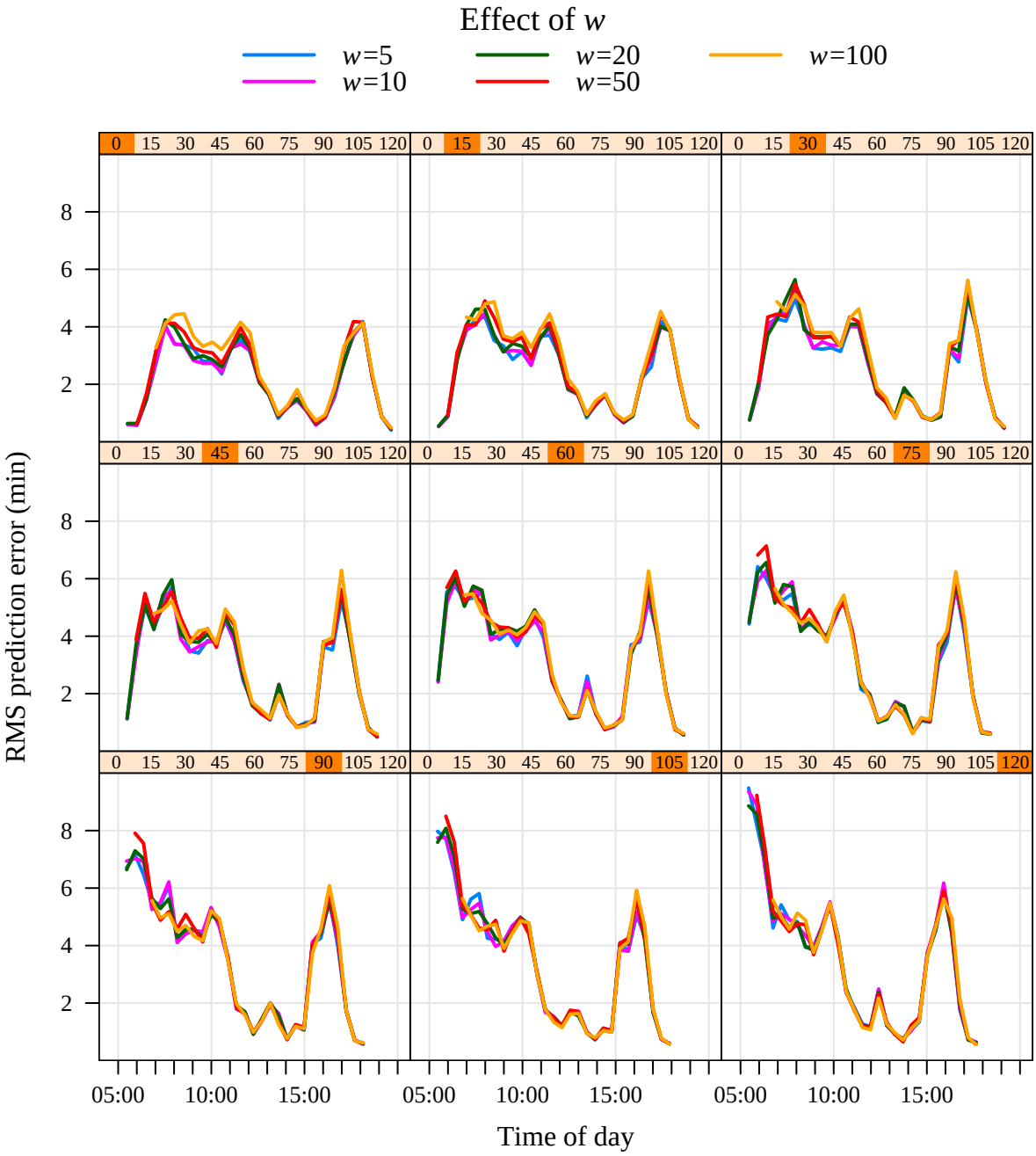


Figure 18: This figure uses the $m_2(\cdot)$ distance function and looks at the effect of the window size parameter w. The prediction error is minimal when w is around 20. The value of k was held fixed at 4.

A Summary of phase one: data organization

This appendix briefly summarizes the work in phase one of this project concerning the organization of a MIDAS data repository. A full description of the work is given in [1].

A.1 The MIDAS TCD format

Loop detectors are positioned in each lane of the carriageway at a given geographical position on the road network. Each day's worth of per minute data from a given site is aggregated across a set of sites at a *control office* (CO). While there may be only tens of CO's, each one may handle data from several hundred loop detector sites. The data for all the sites at a given CO for a given day is assembled into a single TCD file. The TCD file format consists of a single flat file of binary data that lists in unspecified order the data for each site separated by small amounts of header information giving the site's address and the number of lanes. The individual measured values are stored as either one byte or two byte values and are therefore reasonably compact. In summary, a TCD file is generated for each (CO,date) pair and a typical file size may be in the range of 5–10 MB.

For the TCD format, extracting data for a given subset of sites is not straightforward as there is no way, other than by reading through the file, to select portions of data for particular sites. This became an obstacle in some earlier attempts by one of us (RJG) to use the MIDAS data in the TCD format for high throughput applications.

A.2 Revised storage formats to support random access

In phase one of this project we have converted the data stored in TCD format into an alternative format, based on the popular ZIP file format (such files commonly have the `.zip` file extension). This archive file format includes a directory section implementing a lookup mechanism between an archive member name and the byte offset within the file where that member is stored together with its length (other metadata is also stored for each archive member relating to dates, ownerships and miscellaneous comments). ZIP files can be manipulated using both command-line tools (`zip` and `unzip`) which allow insertion and extraction of member files as well as through readily available software libraries which support similar facilities for use within programming languages.

Other common file archive formats, notably the `tar` format commonly used as a tape storage format on Unix systems, lack such a directory and in essence provide only sequential access to archive members in the same manner as the TCD format. Thus, the ZIP file format seemed a natural first choice to consider in place of the TCD format. It has the advantages that it is easily manipulated with commonly available tools and libraries and, importantly, allows random access to the data which is necessary for high throughput performance.

A tool was written in Java (using the `java.util.zip` standard Java library package) to read TCD files and convert them to a ZIP file based storage layout. A single ZIP file was con-

structed for each TCD file in a repository of MIDAS TCD files. Thus, each ZIP file contained the data for a given (CO,date) pair. Within the ZIP file, the data for different sites was stored with each variable corresponding to a different archive member. Thus a day's worth of measured values could be directly extracted for any given variable at any site within the CO. The content of the values remained identical to the binary representation used in the TCD format. Additional metadata was constructed in a archive member for each site specifying details such as the number of lanes and another archive member was provided to hold metadata pertaining to the entire ZIP file. In the future, such metadata might include mappings between site addresses and a variety of geographical location information and details of any missing or corrupt data.

The repository considered in the experiments described in this report consists of MIDAS traffic data from 1995 till mid-2004 and comprises some 30,000 TCD files occupying 137 GB of disc storage. The Java converter tool ran in about 4 hours constructing an equivalent number of ZIP files occupying some 165 GB of disc storage. The additional storage requirement of the ZIP files is due to the extra directory information saved within each ZIP file. This additional storage represents a price or overhead for faster access to data at such fine levels of granularity.

A.3 Further refinements

Further refinements can be made to the data organization. A new custom directory layout, less elaborate than the one used in ZIP files, could have been designed. However, the small gain in the run time of reading the directory each time the ZIP file is opened and the reduction in storage overhead that would have resulted would have been at the expense of extra inconvenience. In place of standardly available tools and libraries bespoke software would be required.

The file format chosen reflects that used in the TCD files but with the additional feature of random access rather than sequential access. A more radical change of the data layout could, for example, implement a single ZIP file for *each* site and use the archive members to distinguish between data for different days. This would assist applications requiring access to data for different days at the *same* site since the overhead of opening the ZIP file and reading the directory information would be amortized over the accesses to data on different days. A potential disadvantage of this layout is that the insertion of new data over time requires the modification of many ZIP files. In contrast, the layout adopted for our experiments requires just one new ZIP file to be added for each day. ZIP files once created thus remain immutable and this invariant can often help where backup of data is required. In our experiments, backup was not essential since the repository could be re-created within several hours by re-running the converter on the original (and carefully backed-up) TCD data.

The ZIP file format maintains the directory information that implements the name lookup within the file as a simple table. Whenever the ZIP file is opened for reading, this directory is read and a suitable data structure (usually a hash table) is built in memory. Thus the size of the directory, or equivalently the granularity at which data is stored, will affect the speed at which ZIP files can be opened. Smaller directory tables would imply less overhead from opening the ZIP file but yield access to data in more aggregated forms. We have not explored this trade-off

further here though further improvements are certainly possible. Alternative approaches might maintain the hash table itself, or equivalent index, within the file thus avoiding the need to build it each time the ZIP file is opened for reading. The BerkeleyDB library is one such approach which uses the highly effective B-tree data structure to maintain its index.

Our use of ZIP files does not include the use of compression which is an optional featured supported by the tools and libraries standardly available. The use of compression would certainly save some disc space at the expense of additional time to decompress the data whenever it is read.

One particularly radical alternative would be to dispense with file archive formats and just use the file system itself to provide access to data by file directory and file name. However, taking a data repository of the size being considered here would require many millions of separate files (one per day per site per variable, say) and quickly run into operating system constraints that limit the total numbers of files. Although, specially configured file systems can be constructed to handle this situation this would add significant additional burdens to use of the data.

The phase one report [1] describes a series of benchmark experiments to investigate running times for a programme to extract MIDAS data in the varying formats and using tools implemented in both Python (an interpreted scripting language) and Java. Full results are given in the phase one report but the experiments clearly demonstrated the value of augmenting a MIDAS repository with an indexing capability to support efficient random access to a fine grain level of data.