Preliminary Report

# LOOP IMPUTATION OF HOURLY DATA ON A SECTION OF THE I-5 (BETWEEN ROUTES 14 AND 99) VIA FUNCTIONAL PRINCIPAL COMPONENT ANALYSIS

JAN DE LEEUW, IRINA KUKUYEVA

Abstract. We present the results of the Functional Principal Component Analysis and Imputation on hourly loop counts on a section of the Interstate-5 freeway (between Routes 14 and 99).

## 1. Data Overview: North Direction

Total observations for each intersection: 96,432
Missing Percentages:

- Route 126 - 8.82%
- Hungry Valley - 30.1%
- Wheeler Ridge - 29.9%
- Route 14 - 70.1%

### 1.1. **Previous Findings.**

As there is about 35% missing data for the North direction of traffic flow, we have previously tried to impute the counts via Least Squares (long time series with indicator variables for day of the week, month, year and hour). The resulting loop counts were too smoothed to be of much use. Therefore, we looked for alternative methods of imputation.

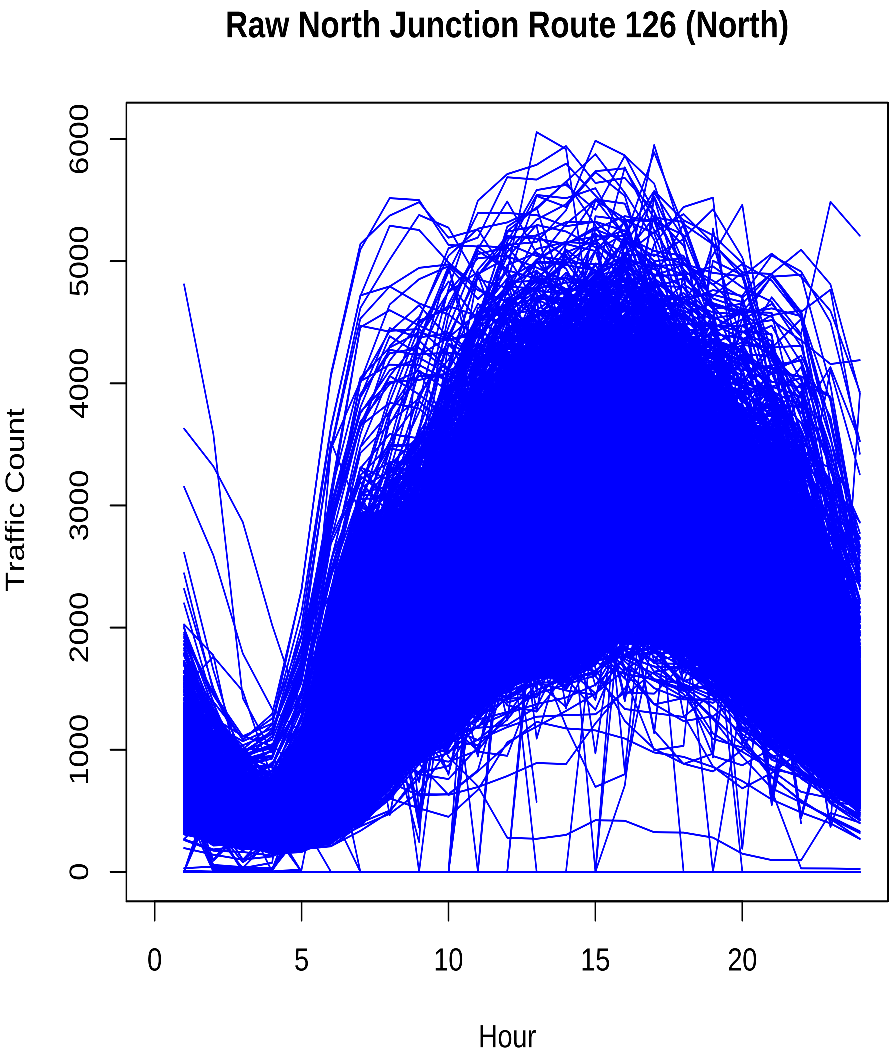
One such method is to use Functional Principal Component Analysis, which takes advantage of the fact that traffic looks similar each 24-hour period. As a result, each intersection was converted to have each separate day as a row in the data matrix and each hour of the day to be a column. The resulting dataset is 4018 by 24. (Please see the Appendix for more information.)

*Date*: January 27, 2008.

*Key words and phrases.* Traffic, FPCA, Imputation.

## 2. Visualizing the Data, One Intersection at a Time

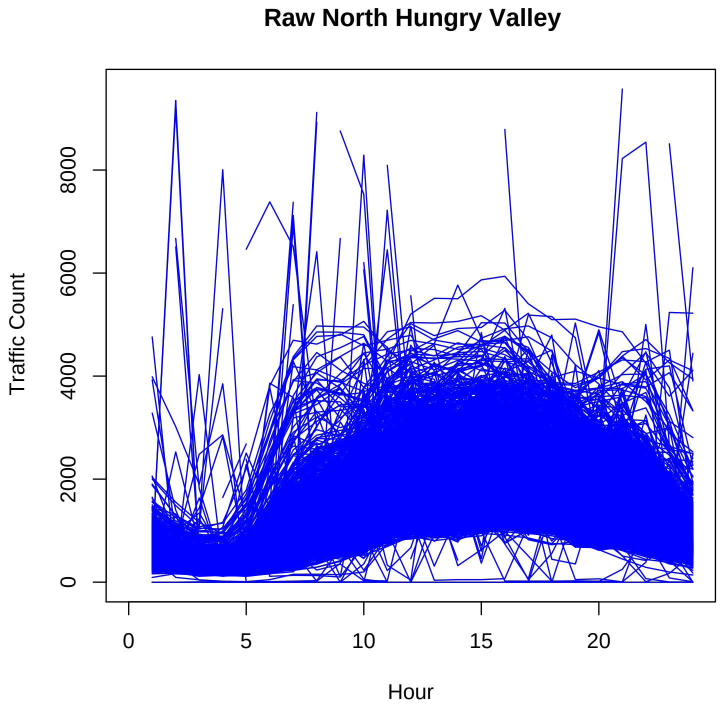
2.1. **Route 126.** In Figure 1 we plot the raw loop counts for each hour for Route 126. A distinctive trend emerges that FPCA will try to model. In addition, we can see that some missing values are zeros in the dataset.



Rte126n.pdf

Figure 1. Raw Hourly Loop Counts on Route 126, North Junction, Between Routes 14 and 99

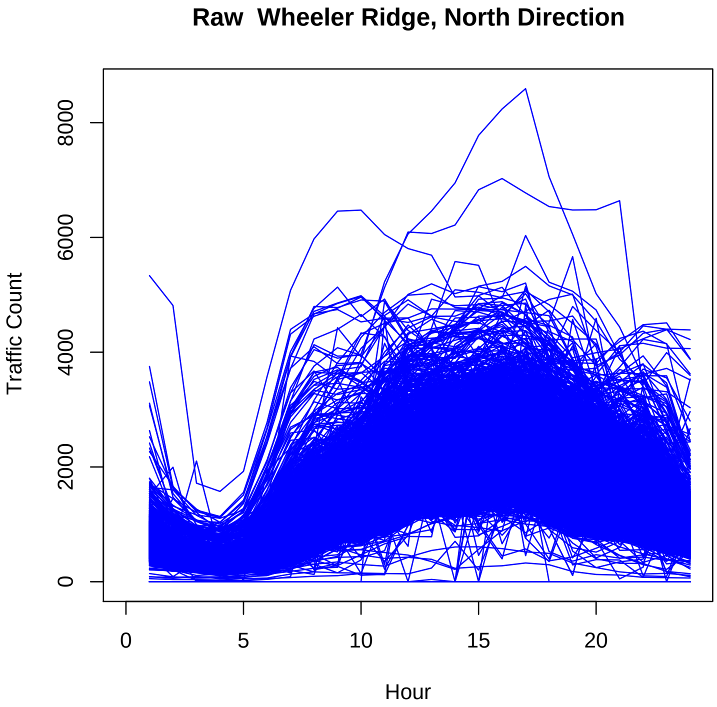
2.2. **Hungry Valley.** We plot the raw loop counts for each hour for Hungry Valley in Figure 2. A distinctive trend emerges, but not quite as pronounced as for Route 126. This may be a result of missing data. As a result, we expect FPCA to model this accordingly. In addition, we can see that, as in the previous dataset, some missing values are coded as zeros in the dataset.



Hungryn.pdf

FIGURE 2. Raw Hourly Loop Counts on Hungry Valley, North Direction, Between Routes 14 and 99

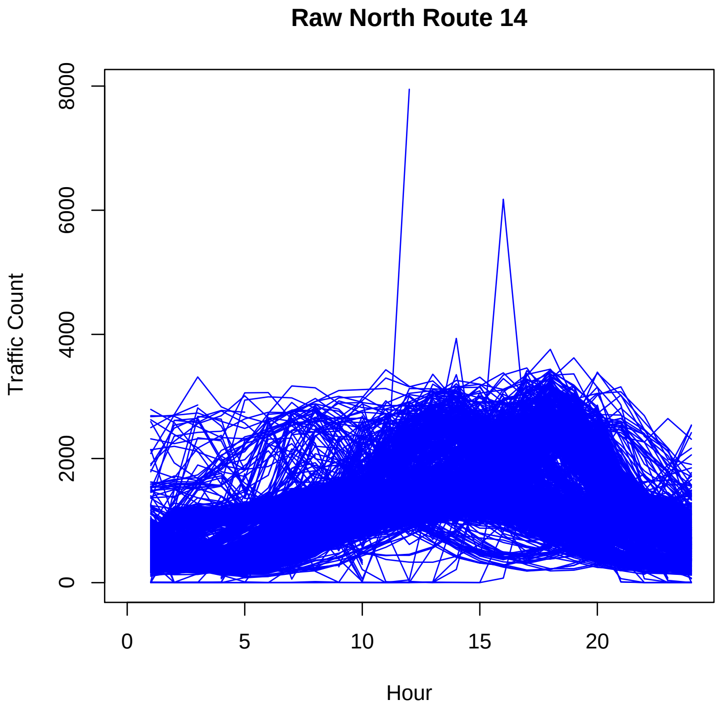
2.3. **Wheeler Ridge.** Figure 3 shows the raw loop counts for each hour for Wheeler Ridge. Similarly, there is a distinctive trend for this intersection as well. Also, we can see that, as in the previous dataset, some missing values are coded as zeros in the dataset.



Wheelern.pdf

Figure 3. Raw Hourly Loop Counts on Wheeler Ridge, North Direction, Between Routes 14 and 99

2.4. **Route 14.** We plot the raw loop counts for each hour for Route 14 in Figure 4. We can see that there is a trend in the dataset, but it the least pronounced of all the intersections (and more smooth). This may be a result of missing data, as it is as high as 70%. As a result, we expect FPCA to model this accordingly. In addition, we can see that, as in the previous datasets, some missing values are coded as zeros here, which only adds to the smoothness we observe.



Rte14n.pdf

Figure 4. Raw Hourly Loop Counts on Route 14, North Direction, Between Routes 14 and 99

## 3. Imputation Results, One Intersection at a Time

3.1. **Route 126, North Direction.** We use one of the imputation methods illustrated in the *Curves paper, such as the function SVDCenter. The algorithm (almost) converged in 50,000 iterations, with Loss=2,482,285,267.899635 (which took over 2 hours to run). When we proceeded to further impute on the resulting dataset, the algorithm converged in 3 iterations with a Loss of zero. As a result, we will run the algorithm again with a larger number of iterations to double check our findings.*

*Next, to check the accuracy of our method, we extracted the fitted values for an arbitrary day for Route 126 and compared them with the actual values (for a day where there were no missing values). We present the results in Figure 5, with the original values in blue. We can see that the fit follows the trends in traffic quite accurately.*

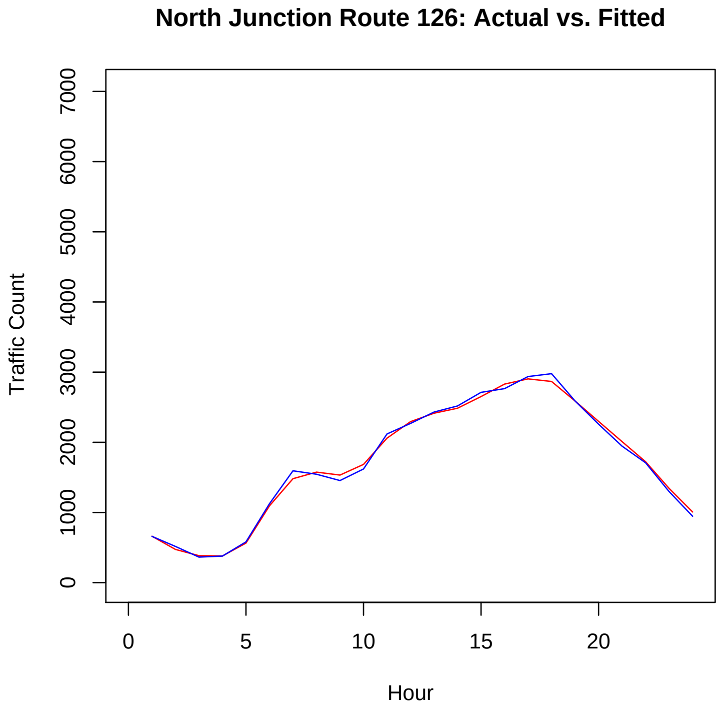
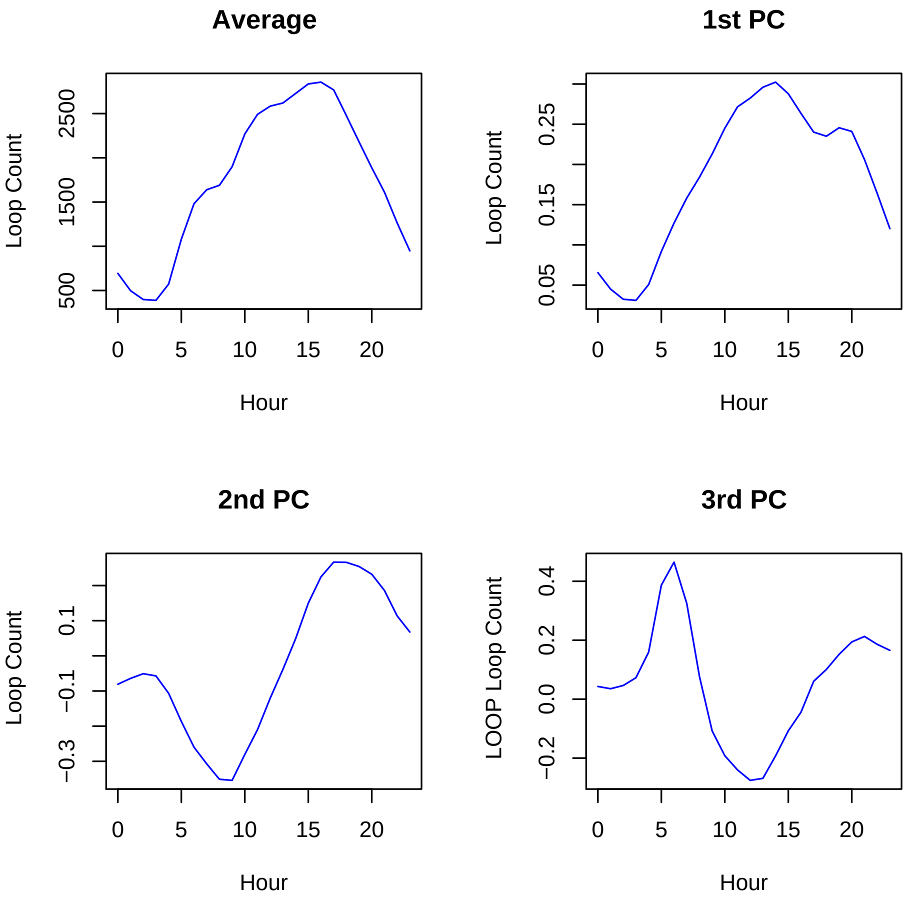


FIGURE 5. Comparison of Imputation Method for Route 126, North Direction, Between Routes 14 and 99

*In addition, we present the Average along with the first three Principal Curves (Figure 6) in order to better understand the trend in the imputed dataset. We can see that the Average and 1st Principal Curve are very similar. They show that traffic is lowest during the early morning hours and increases steadily throughout the day, with a peak between 3 and 6 PM; then it decreases.*

*The 2nd Principal Curve might be interpreted as the rate of change of traffic: decreases slowly until about 3AM (either congested - not likely - or virtually no traffic), then increases steadily through 9AM (more cars on the road), only to increase rapidly until 6PM (as more and more people get out of work), then level off (freeway is again, either congested - not likely - or virtually no traffic).*

*Then the 3rd Principal Curve might talk about an individual car's 'acceleration' at this intersection; that is, cars go fast until about 6AM (when they go the fastest), where they begin to go slower than at 6AM, but fast nonetheless. After 12PM, the road becomes more congested and cars do not drive as fast relative to each other anymore.*



*PCs.pdf*

Figure 6. Principal Component Curves from Imputation Method for Route 126, North Direction, Between Routes 14 and 99

## 4. R Code

```
# opened "countsn FPCA.RData"
# want to impute on the missing ones from raw data file
# change directories
setwd("C:/Documents␣and␣Settings/Irina/My␣Documents/Irina/I−5␣Traffic␣
    Research/Loop␣Imputation/FPCA␣on␣Counts")
# create new dataset with only the four relevant columns (
    intersections)
data2=data[, −c(1:6)]
# dropped milepost and kern−line
detach()
attach(data2)
names(data2)
```

```
summary(data2)  # only the four intersections

data3<-t(data2)
detach()
as.data.frame(data3)

# run the following to store the function
imputeMat<-function(mat, fitme, eps=1e-6,niter =100,verbose=TRUE, pars=NULL) {
n<-nrow(mat) ; m<-ncol(mat) ; oloss<-Inf ; iter<-1
fitted<-matrix (0 ,n,m) ; imputed<-mat
repeat {
for ( i in 1 :n) {
ind<-which( is.na(mat [ i , ] ) )
imputed[ i , ind ]<-fitted [ i , ind ]
}
nloss<-sum( ( imputed-fitted) ^2)
motor<-fitme ( imputed , pars ) ; fitted<-motor\$fitted ; extra<-motor\$extra
if ( verbose )
cat ( "␣Iteration␣:␣" , formatC( iter , digits =6,width=6) ,
"␣Loss:␣" , formatC( oloss , digits =6,width=12,format="f" ) , "␣==>" , formatC( nloss , digits =6,width=12,format="f" ) , "␣\n" )
if ( ( ( oloss-nloss ) < eps ) || ( iter == niter ) ) break( ) ;
oloss<-nloss ; iter<-iter+1
}
return( list( iter=iter , loss=nloss , fitted=fitted , extra=extra ) )
}

# In the Curves paper, compared among the smoothed curves (in blue) to determine
# which process to use to impute missing data
# settled on using SVDCenter

fitSVDCenter<-function (mat , pars ) {
n<-nrow(mat) ; av<-as.vector ( apply (mat,2 ,mean) ) ; mav<-outer ( rep(1 ,n), av )
sv<-svd(mat-mav,nu=pars , nv=pars )
return ( list ( fitted=mav+tcrossprod ( sv$u , ( sv$v )%*%diag ( sv$d[
      1: pars ] ) ) ) ,
extra= list ( av , sv ) ) )
}

# Rte 126
# want matrix with hour as columns
attach(data2)
rte126n=matrix(0, 4018,24)        # initialize to zero with appropriate
    dimensions
# 24 columns, one for each hour
m=24
i=1
j=1
repeat{
rte126n[j,]<-t(n_jct_rte126[i:m])
i<-i+24
            m<-m+24
            j<-j+1
            if(m>96432)break()
}
# to check that entered correctly:
n_jct_rte126[1:48]

# adding time of day for each column
colnames(rte126n)<-c("hr0", "hr1", "hr2", "hr3", "hr4", "hr5", "hr6",
    "hr7", "hr8", "hr9", "hr10", "hr11", "hr12", "hr13", "hr14", "hr15"
    , "hr16", "hr17", "hr18", "hr19", "hr20", "hr21", "hr22", "hr23")

# to check:
rte126n[1:10,]

# plot of raw data:
pdf( "RAW_Rte126n.pdf" )
```

```
plot (0:24 , seq(0 ,max(rte126n, na.rm=TRUE) , length=25) , type="n" , xlab="Hour" , ylab="Traffic␣Count" , main="Raw␣North␣Junction␣Route␣126␣(North)")
for ( i in 1 :4018 ) lines(rte126n[i,], col="BLUE")
dev.off()

# imputing data
# which rows have less than 5 missing values
# indx_126<-which( apply( rte126n ,1 , function( x ) length(which( is.na( x ) ) ) ) <5)
# above works if we want to smooth the data, not impute
# when smoothing, the algorithm converges in 26 iteration
# Loss= 2467201444.035013

# to impute, we want to select all the observations:
# indx=1:4018
# did not converge in 500 iterations
rte126svd<-imputeMat( rte126n, fitSVDCenter , pars=3, niter =50000)
# did not converge in 50,000 iterations
# soĚ saved this and let it impute again
rte126svd2<-imputeMat( rte126svd\$fitted, fitSVDCenter , pars=3, niter =50000)

# plot of the imputed data
pdf( "Route126n_fitted.pdf" )
plot (0:24 , seq(0 ,max(rte126svd\$fitted, na.rm=TRUE) , length=25) , type="n" , xlab="Hour" , ylab="Traffic␣Count" , main="Smoothed␣North␣Junction␣Route␣126")
for ( i in 1 :4018 ) lines(rte126svd\$fitted[i,], col="BLUE")
dev.off()

# comparing an arbitrary hour without missing data to that of imputed data
# stores which rows have no missing data
ind126<-which( apply( rte126n ,1 , function( x ) length(which( is.na( x ) ) ) ) ==0)
set.seed(1282008)
```

```
i126=sample(ind126,1); i126

pdf( "Route126n_Comparison.pdf" )
plot (0:24 , seq(0 ,max(rte126svd\$fitted, na.rm=TRUE) , length=25) ,
    type="n" , xlab="Hour" , ylab="Traffic Count" , main="North 
    Junction Route 126: Actual vs. Fitted")
lines(rte126svd\$fitted[i126,], col="red")
lines(rte126n[i126,], col="blue")          # original
dev.off()
# similarly for the other three intersections

# graphing Principal Curves, using Method from Figure 17 in Curves
pdf( "Route126n PCs.pdf" )
par(mfrow=c(2,2))
plot(0:23,rte126svd\$extra[[1]],type="l",col="BLUE",xlab=" Hour ",ylab
    ="Loop Count", main="Average")
# avg
plot(0:23,rte126svd\$extra[[2]]\$v[,1],type="l",col="BLUE",xlab=" Hour
    ",ylab=" Loop Count", main="1st PC")
# 1st
plot(0:23,rte126svd\$extra[[2]]\$v[,2],type="l",col="BLUE",xlab=" Hour
    ",ylab=" Loop Count", main="2nd PC")
#2nd
plot(0:23,rte126svd\$extra[[2]]\$v[,3],type="l",col="BLUE",xlab=" Hour
    ",ylab=" LOOP Loop Count", main="3rd PC")
# 3rd curve
dev.off()
####
```

*E-mail address:* `deleeuw@stat.ucla.edu, ikukuyeva@stat.ucla.edu`