MESgenCov: An R Package for generating covariance matrices from precipitation chemistry data

Hessa Al-Thani[1] Jon Lee[2]

Dept. of Industrial and Operations Engineering, Univ. of Michigan, Ann Arbor, MI 48105 USA

[1]hessakh@umich.edu [2]jonxlee@umich.edu

MESgenCov 0.1.0

Abstract

We present an R package for temporally fitting multivariate precipitation chemistry data and extracting a covariance matrix for use in MES (maximum-entropy sampling). We provide multiple functionalities for modeling and model assessment. The package uses data from the NADP/NTN (National Atmospheric Deposition Program / National Trends Network) on their set of more than 370 monitoring sites, 1978–present. The user specifies the sites, chemicals and time period desired, fits a user-defined univariate model for each site and chemical selected, and produces a covariance matrix for use by MES algorithms.

1 The MES problem

The MES (maximum-entropy sampling) problem (see [SW87,SW00,FL00,Lee12]) has been applied to many domains where the objective is to determine a most informative subset S, of presepcified size s, from a set of n Gaussian variables N. This is done by seeking to maximize the (log) determinant of the covariance matrix for some $S \subseteq N$ with $|S| = s$ (see [KLQ95,Lee98,AFLW99,LW03,HLW01,AL04,BL07,Ans18b,Ans18a]). A key area of application has been in environmental montoring (see [ZSL00,BLZ94,GLSZ93], for example). In this R package, we use precipitation chemistry data (ammonium, calcium, chloride, hydrogen, magnesium, nitrate, pH, potassium, sodium, and sulfate) gathered by the NADP (National Atmospheric Deposition Program) at over 370 sites across the U.S.A. (see [NAD18]). For these instances of the MES problem, n *user-specified* site/chemical pairs comprise N.

2 NADP data description

The NADP maintains the NTN (National Trends Network); this network measures the chemistry of precipitation at monitoring sites across the U.S.A. Our R package makes use of the weekly data measuring mg/L of chemicals, such as sulfates, in collected precipitation. We also use the daily data measuring precipitation at each site. Both datasets are available in our package and can be loaded respectively as

```
data("weeklyConc")
data("preDaily")
```

A full description of the data can be obtained at NADP/NTN weekly meta. For a full description of the daily precipitation data, see NADP/NTN daily meta. Small snapshots of the data can also be viewed using:

```
  weeklyConc[1:5,1:5]

##   siteID              dateon             dateoff yrmonth    ph
## 1   AB32 2016-09-13 18:40:00 2016-09-20 15:10:00  201609 -9.00
## 2   AB32 2016-09-20 15:15:00 2016-09-28 16:00:00  201609 -9.00
## 3   AB32 2016-09-28 16:00:00 2016-10-05 16:55:00  201610  6.56
## 4   AB32 2016-10-05 16:55:00 2016-10-11 17:00:00  201610 -9.00
## 5   AB32 2016-10-11 17:00:00 2016-10-18 20:00:00  201610 -9.00
```

This outputs the first 6 rows, first 4 columns and 9th column of the weekly raw data, columns 5 to 8 contain data that isn't relevant to our analysis.

3 MESgenCov implementation

The **MESgenCov** package contains functions in the `S3` class to create a covariance matrix from the desired subset of NADP data. The function `getCov()` returns a covariance matrix, a list of univariate model summaries, and a table of normality tests produced by the MVN R package. The covariance matrix is produced from a subset of the NADP/NTN data that is specified by the user. For sites with missing data, `getCov()` will fill in predicted values based on the univariate model for each site (see section 3.1). To avoid sites with a small sample size for the specified time-frame, the function `getSites()` outputs a vector of the sites with the largest sample of data for a given time-frame and measured chemical (see section 3.2). To find sites that are spatially "spread out" but have at least some specified sample size, the function `maxDistSites()` can be used (see section 3.2).

3.1 getCov

3.1.1 Input

`getCov()` takes a 15 column data frame as input where each column corresponds to one of the following user-specifications, given in the table below.The 15 specifications in the input allow the user to specify the subset of data to analyze and gives the user options in displaying different parts of the analysis.

Arguments	Definition
weeklyB	TRUE if weekly data should be analyzed and FALSE if monthly data should be analyzed
startdateStr	Date and time of when to start analyzing the data, in the format = "m/d/y H:M"
enddateStr	Date and time of when to stop analyzing the data, in the format = "m/d/y H:M"
comp	Vector of strings of pollutants or acidity levels to be analyzed, the pollutants name should be used as it appears in weeklyCSV
use36	TRUE if default 36 sites should be added, FALSE otherwise
siteAdd	List of strings of siteIDs that should be analyzed
outlierDatesbySite	List of sites where outliers should be analyzed
siteOutliers	List of sites where outliers should be removed
removeOutliers	Specify siteID string for outlier analysis
plotMulti	TRUE if multivariate analysis plots should be displayed, FALSE otherwise
sitePlot	Specify siteID to be plotted
plotAll	TRUE if plots for all sites should be displayed, FALSE otherwise
writeMat	TRUE if .mat file of the resulting covariance matrix should be written in the working directory
seas	Approximate periodicity of data, typically 12 for monthly data and 52 for weekly data
r	Integer <=5, see univariate model
k	Integer <= 5, see univariate model

A default set of inputs can be found in the internally stored dataframe "defaultInput". After storing it in a variable in the user's workspace the input can be changed in the following way:

```
data("defaultInput")
df <- defaultInput
df

##   weeklyB   startdateStr     enddateStr comp use36 siteAdd
## 1   FALSE 01/01/83 00:00 12/31/86 00:00  SO4  TRUE    NULL
##   outlierDatesbySite siteOutliers removeOutliers plotMulti sitePlot
## 1               NULL         NULL           NULL     FALSE     NULL
##   plotAll writeMat seas r k
## 1   FALSE    FALSE   12 1 1

df$enddateStr   <- "12/31/88 00:00"
```

3.1.2 Output

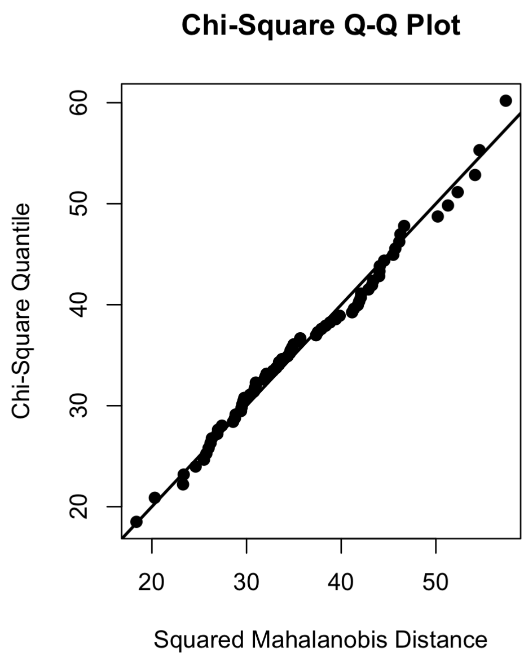
The function `getCov` produces the following outputs:

Output	Definition
cov	Covariance matrix produced by univariate model residuals
listMod	List of univariate model summaries produced by lm()
sites	List of sites that were analyzed
mvn	Output of the MVN package
univariateTest	univariate test output, also by the MVN package
residualData	Data frame of residuals produced by the univariate model
residualDataNA	Data frame of residuals, where missing values are ledt as NA
rosnerTest	Output of the rosner test for outlier analysis produced by the EnvStats package
pred	List of predicted values produced by the univariate model for each site
noutliers	Number of outliers detected by the outlier analysis, default is 0 if no sites are analyzed for outliers

Here we present some examples of the various outputs.

1. Multivariate and univariate normality

```
df$plotMulti    <- TRUE
df$k            <- 3
g <-getCov(df)
```



```
g$univariateTest[1:5,]
```

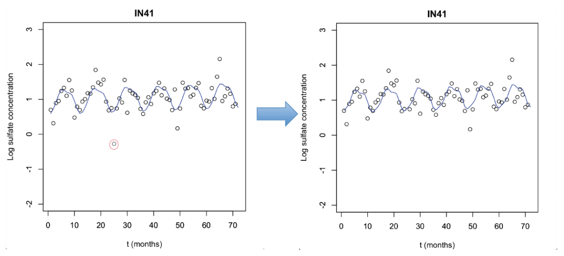
```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk  AL10SO4     0.9347   0.001      NO
## 2 Shapiro-Wilk  IL11SO4     0.9894  0.8121      YES
## 3 Shapiro-Wilk  IL18SO4     0.9909  0.8854      YES
## 4 Shapiro-Wilk  IL19SO4     0.9183   2e-04      NO
## 5 Shapiro-Wilk  IL35SO4     0.8709  <0.001      NO
```

2. Outlier test for specific tests

```
df$siteOutliers <- list(c("IN41"))
df$sitePlot     <- "IN41"
g <- getCov(df)
i <- match("IN41",g$sites)
g$rosnerTest[[i]]$all.stats

##   i       Mean.i      SD.i      Value Obs.Num    R.i+1 lambda.i+1 Outlier
## 1 0 -0.006929523 0.2813805 -0.9358517      25 3.301302   3.267957    TRUE
## 2 1  0.006153889 0.2603946 -0.7193545      30 2.786188   3.262821   FALSE
## 3 2  0.016518294 0.2470843  0.7215347      66 2.853344   3.257596   FALSE
```

```
df$outlierDatesbySite <- c("IN41",25)
getCov(df)
```

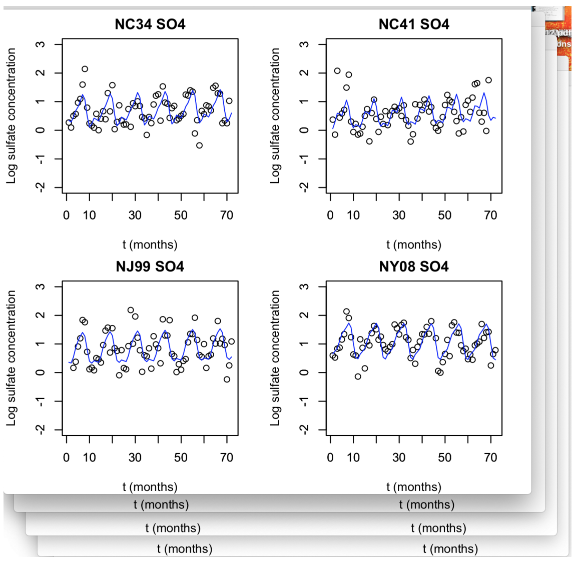


For all sites,

```
df$siteOutliers   <- list(g$sites)
df$removeOutliers <- list(g$sites)
g <- getCov(df)
```

3. Plot all sites

```
df$plotAll <- TRUE
getCov(df)
```



4. Covariance matrix

```
  df$use36                <- FALSE
  df$siteAdd              <- list(c("OH71", "WV18", "MI53"))
  df$sitePlot             <- NULL
  df$siteOutliers         <- NULL
  df$outlierDatesbySite <- NULL
  df$removeOutliers       <- NULL
  g                       <- getCov(df)
  round(g$cov,digits = 4)

##          OH71SO4 WV18SO4 MI53SO4
## OH71SO4  0.0770  0.0231  0.0092
## WV18SO4  0.0231  0.0911  0.0124
## MI53SO4  0.0092  0.0124  0.1096
```

5. Save covariance matrix as .mat file

   This is done by simply setting the 5th last input to TRUE then the .mat file will be saved to the user's current working directory.

```
  df$writeMat <- TRUE
  g           <- getCov(df)
```

   In the case that the user has already generated an output by the function `getCov()`, it is possible to also create the mat file in the following way.

```
  library(rmatio)
  write.mat(g$cov,filename = "covariance1.mat")
```

6. Univariate model summaries

```
  sites  <- g$sites
  i = match(c("OH71"),sites)
  g$listMod[i]
```

```
## [[1]]
##
## Call:
## lm(formula = y1 ~ I(cos(t * (2 * pi/seas))^p) + I(sin(t * (2 *
##     pi/seas))^p) + I(cos(t * (2 * pi/seas) * 2)^p) + I(sin(t *
##     (2 * pi/seas) * 2)^p) + I(cos(t * (2 * pi/seas) * 3)^p) +
##     I(sin(t * (2 * pi/seas) * 3)^p) + I(t), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52359 -0.16772 -0.01885  0.18184  0.90868
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.1110107  0.0702653  15.812  < 2e-16 ***
## I(cos(t * (2 * pi/seas))^p)    -0.3541145  0.0487427  -7.265 6.37e-10 ***
## I(sin(t * (2 * pi/seas))^p)    -0.1108821  0.0491147  -2.258   0.0274 *
## I(cos(t * (2 * pi/seas) * 2)^p)  0.0499868  0.0487427   1.026   0.3090
## I(sin(t * (2 * pi/seas) * 2)^p)  0.0796743  0.0488005   1.633   0.1075
## I(cos(t * (2 * pi/seas) * 3)^p)  0.0390617  0.0487427   0.801   0.4259
## I(sin(t * (2 * pi/seas) * 3)^p)  0.0535530  0.0487427   1.099   0.2760
## I(t)                            0.0009172  0.0016779   0.547   0.5865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2923 on 64 degrees of freedom
## Multiple R-squared:  0.4995,Adjusted R-squared:  0.4448
## F-statistic: 9.125 on 7 and 64 DF,  p-value: 9.097e-08
```

7. Output all MVN package analysis

   The following output is a call to the MVN package that produces mutivariate analysis based on the Mardia method, univariate analysis based on the Shapiro-Wilson method and an multivariate outlier test that is presented as a plot and not as an output in the user's R console.

```
  g$mvn

## $multivariateNormality
##              Test        Statistic             p value Result
## 1 Mardia Skewness  24.362552007168  0.0066940769960186     NO
## 2 Mardia Kurtosis 2.59883767062138 0.00935399956866512     NO
## 3             MVN             <NA>                <NA>     NO
##
## $univariateNormality
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk  OH71SO4     0.9593    0.0201    NO
## 2 Shapiro-Wilk  WV18SO4     0.9823    0.4073    YES
## 3 Shapiro-Wilk  MI53SO4     0.9487    0.0053    NO
##
## $Descriptives
##          n          Mean   Std.Dev       Median        Min       Max
## OH71SO4 72 -1.025964e-17 0.2775010 -0.018850157 -0.5235855 0.9086803
## WV18SO4 72 -6.613633e-18 0.3018523  0.003571524 -0.7192734 0.7195297
## MI53SO4 72 -1.356006e-18 0.3310569 -0.012777134 -0.5260723 1.2265622
##               25th      75th      Skew   Kurtosis
## OH71SO4 -0.1677203 0.1818372 0.6819771 1.10314285
## WV18SO4 -0.1868506 0.1776964 0.2203785 0.05048396
## MI53SO4 -0.2596244 0.2318249 0.7628797 0.88211091
##
## $multivariateOutliers
##   Observation Mahalanobis Distance Outlier
## 1           1               16.524    TRUE
## 2           2               16.144    TRUE
## 3           3               15.684    TRUE
## 4           4               15.421    TRUE
## 5           5               10.627    TRUE
## 6           6               10.616    TRUE
## 7           7                9.477    TRUE
```

The specific call the the MVN package is,

```
mvn(dfRes[,-1], subset = NULL, mvnTest = "mardia", covariance = TRUE,
    tol = 1e-25, alpha = 0.5, scale = FALSE, desc = TRUE, transform = "none",
    univariateTest = "SW",  univariatePlot = "none", multivariatePlot ="none",
    multivariateOutlierMethod = "none", bc = FALSE, bcType = "rounded",
    showOutliers = FALSE, showNewData = FALSE)
```

See [KGZ14] for details on the MVN package.

8. Output dataframe of residuals

```
g$residualData[1:5,]
```

```
##      OH71SO4     WV18SO4    MI53SO4
## 1 -0.1863739 -0.17114817  0.4782343
## 2 -0.2965845 -0.15398625  0.2875634
## 3 -0.4194283 -0.02188007 -0.5112095
## 4 -0.1446214 -0.49188947 -0.3921147
## 5 -0.5235855 -0.53657443  0.0579487
```

3.2 Functions for getting a vector of sites

`getCov()` takes site lists as input, the function `getSites()` produces a list of sites with available data for a specified time frame. The code below produces a list of 36 sites with the most weekly data between the years 1983-1986.

```
result <- getSites("01/01/83 00:00", "12/31/86 00:00",36,104,"SO4","")
result$finalList
```

```
##  [1] "OH71" "NY08" "WV18" "MI53" "NH02" "OH49" "PA42" "ME09" "IN34" "MA13"
## [11] "NY52" "NY10" "WA14" "NY20" "OH17" "ME00" "TN00" "IL63" "MI99" "WI28"
## [21] "IN41" "PA29" "WI36" "ME02" "MI09" "MO05" "NC03" "NJ99" "PA15" "CO19"
## [31] "MN18" "WI37" "AR27" "KS31" "ME98" "MO03"
```

The 4th input specifies the minimum sample of weekly data required to be included in the produced list and the last input tells the function to only look at sites in the North Eastern region of the US. Other options for region "W","S","N", see Appendix A for the precise geographic split.

```
  NSites <- getSites("01/01/83 00:00", "12/31/86 00:00",36,104,"SO4","N")
  NSites$finalList

##  [1] "OH71" "NY08" "MI53" "NH02" "OH49" "PA42" "ME09" "IN34" "MA13" "NY52"
## [11] "NY10" "NY20" "OH17" "ME00" "MI99" "WI28" "IN41" "PA29" "WI36" "ME02"
## [21] "MI09" "NJ99" "PA15" "MN18" "WI37" "ME98" "IL11" "IL18" "MN16" "MI26"
## [31] "NE15" "VT01" "NY99" "MA01" "MA08" "MN27"
```

The function `maxDistSites()` prioritizes sites that are farther away from each other.

```
  maxdist <- maxDistSites("01/01/83 00:00", "12/31/86 00:00",36,104,"SO4",1)
  maxdist$finalList

##  [1] "OH71" "WA14" "TX04" "FL11" "ME00" "WY06" "MN27" "LA12" "CA45" "OK00"
## [11] "NY99" "GA41" "MI99" "AZ03" "MT05" "NC35" "MO05" "CO00" "WY99" "IN34"
## [21] "KY03" "MI09" "FL03" "MA01" "OR10" "PA42" "AR27" "MN16" "TX21" "VT99"
## [31] "NE15" "VA13" "CO15" "CO22" "NY52" "AR02" "WI28"
```

3.3 Lambert's W transformation on univariate data

For a number of sites the resdiuals produced by the deterministic model have skewed distributions with heavy tails. In particular, this is the case for many sites when the sample of data is taken over a period longer than 4 years. To deal with this issue we've incorporated functions from the LambertW package (see [Goe16])in the function `lambertWtransform()` that will allow a user to transform the residuals produced by the deterministic univariate model. The LambertW package estimates the parameters that fit a LamberW distribution on the given univariate data. Then the underlying gaussian distribution implied by the LambertW distribution is extracted and is used for the multivariate analysis in the function `lambertWtransform()`. The `lambertWtransform()` function takes the following as input: a dataframe of residuals, and two binary inputs specifying whether to plot the multivariate qq plot and whether to produce the mat file containing the covariance matrix with the Lambert W transformed residuals. Details on the algorithms that perform the transformation can be found in [Goe11].

```
  data("dfRes50")
  loutput <- lambertWtransform(dfRes = dfRes50, plotMulti = FALSE, writeMat = FALSE)
  loutput$mvn$multivariateNormality

##              Test         Statistic           p value Result
## 1 Mardia Skewness  22174.3239138031  0.36080892676398    YES
## 2 Mardia Kurtosis -1.52927275390506 0.126196841108458    YES
## 3             MVN              <NA>              <NA>    YES
```

This function will produce a list of four elements that can be called using the following names:

```
data("dfRes50")
loutput <- lambertWtransform(dfRes = dfRes50, plotMulti = FALSE, writeMat = FALSE)
loutput$mvn$multivariateNormality

##              Test         Statistic           p value Result
## 1 Mardia Skewness  22174.3239138031  0.36080892676398    YES
## 2 Mardia Kurtosis -1.52927275390506 0.126196841108458    YES
## 3             MVN              <NA>              <NA>    YES

round(loutput$cov[1:5,1:5],digits = 4)

##         WV18SO4 AK03SO4 CA75SO4 PR20SO4 ND11SO4
## WV18SO4  0.0580 -0.0009  0.0008 -0.0081 -0.0003
## AK03SO4 -0.0009  0.2779 -0.0376  0.0061 -0.0152
## CA75SO4  0.0008 -0.0376  0.3516  0.0080 -0.0640
## PR20SO4 -0.0081  0.0061  0.0080  0.0664 -0.0078
## ND11SO4 -0.0003 -0.0152 -0.0640 -0.0078  0.3235

loutput$newResiduals[1:5,1:5]

##       WV18SO4     AK03SO4    CA75SO4     PR20SO4    ND11SO4
## 1  0.38136269 -0.54050778 -0.2296793 -0.05443715  0.1574453
## 2  0.35080928 -0.46198096  1.4792923 -0.10513708  1.2128039
## 3 -0.14237200  1.09269559 -0.9999155  0.18812756 -0.4768100
## 4  0.47692481  0.86168781  0.5652337 -0.30948198 -0.5916342
## 5 -0.02355482  0.08727396  0.1669882 -0.34881047 -0.2874526

loutput$univariateTest[1:5,]

##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk  WV18SO4     0.9912    0.7109    YES
## 2 Shapiro-Wilk  AK03SO4     0.9933    0.8774    YES
## 3 Shapiro-Wilk  CA75SO4     0.9944    0.9437    YES
## 4 Shapiro-Wilk  PR20SO4     0.9940    0.9199    YES
## 5 Shapiro-Wilk  ND11SO4     0.9933    0.8790    YES
```

1. **loutput$mvn** will show the results of applying the multivariate analysis by the MVN package
2. **loutput$cov** will output the covariance matrix produced by the transformed residuals
3. **loutput$newResiduals** will output the dataframe of Lambert W transformed residuals
4. **loutput$univariateTest** will output the univariate tests produced by the MVN function for the transformed residuals

```
#get list of sites
maxd  <- maxDistSites("01/01/86 00:00","12/31/94 00:00",50,200,"SO4",1)
#create input dataframe
df <- defaultInput

df$siteAdd      <- list(maxd$finalList)
df$startdateStr <- maxd$startDate
df$use36        <- FALSE
df$comp         <- maxd$comp
df$enddateStr   <- maxd$endDate
df$writeMat     <- TRUE
g               <- getCov(df)
g$mvn$multivariateNormality
```

```
##                Test          Statistic              p value Result
## 1 Mardia Skewness  24323.5311757276 2.08544615527066e-05     NO
## 2 Mardia Kurtosis 0.199883303506392    0.841571848809121    YES
## 3             MVN              <NA>                 <NA>     NO

loutput <- lambertWtransform(g$residualData, TRUE,FALSE)
loutput$mvn$multivariateNormality

##                Test          Statistic           p value Result
## 1 Mardia Skewness  23680.5500573711  0.12002749809119    YES
## 2 Mardia Kurtosis -1.44389803480593 0.148767659521509    YES
## 3             MVN              <NA>              <NA>    YES
```

```
indp            <- independenceTest(g$residualData)
indp$test

##   chisq dist likelihood ratio    chisq independent
## 1                   23200.17 1359.182       FALSE
```

4 Methodology

4.1 NADP data processing

We process the raw NADP data in a similar way to earlier uses in the context of the MES problem in the field of environmental statistics. [GLSZ93] analyzes the levels of a chemical's concentration by summing weekly quantities (mg) of the chemical, over a month, and dividing the monthly total by total precipitation (L), over dates in that month, to get monthly values of sulfate concentration (mg/L). For a given monitoring site, chemical, and month $t = 0, 1, \dots, T-1$, let

$$\begin{aligned}
W(t) &:= \text{set of weeks in month } t,\\
D(w) &:= \text{set of days in week } w,\\
c_w &:= \text{recorded chemical concentration (mg/L) for week } w\\
&\quad\ (c_w = * \text{ denotes an unrecorded value}),\\
p_d &:= \text{recorded precipitation quantity (L) for day } d,\\
p_w &:= \text{precipitation quantity (L) for week } w;\ p_w = \textstyle\sum_{d\in D(w)} p_d.
\end{aligned}$$

Then the chemical's concentration (mg/L) for month t is calculated as

$$y(t) := \frac{\sum_{w\in W(t):c_w\neq *} p_w c_w}{\sum_{w\in W(t):c_w\neq *}\sum_{d\in w} p_d}.$$

It should be noted that when there is no weekly value available for the chemical quantity, we do not use the preciptation values for any of the days in such a week (so as to not artificially dilute the chemical concentration level for the month).

Finally, a model is fit to $\log(y(t))$. In [GLSZ93] they use the following model to deseasonalize and detrend the data set of log transformed monthly sulfate concentration values.

$$\log(y(t)) \approx \beta_1 + \beta_2 t + \beta_3 \cos\left(\frac{2\pi t}{12}\right) + \beta_4 \sin\left(\frac{2\pi t}{12}\right). \tag{1}$$

Basically, this is just an affine model $\beta_1 + \beta_2 t$ plus a sinusoidal model with monthly periodicity and intercept β_3.

We found that this model did well in normalizing the error for certain cites but some sites like "MD13" and "NC03" did not do as well. Rather than fix (1) as our model, we provide a more flexible model described in the next section.

4.2 The univariate model

The general model that we provide is

$$\log(y(t)) \approx \sum_{i=0}^{r} \beta_i t^i + \sum_{j=1}^{k} \left[a_j \cos\left(\frac{2\pi jt}{S}\right) + b_j \sin\left(\frac{2\pi jt}{S}\right) \right].$$

The user can specify the degree r for the polynomial part of the model which we think of as a truncated taylor series, aimed at capturing aperiodic trends. Periodic trends are captured via a truncated Fourier series, truncated at level k. The simple model (1) is this one with $r = 1$, $k = 1$ and $S = 12$.

4.3 Internal data sets

```
#sites with maximum distance data sets, get 50 sites
maxd1  <- maxDistSites("01/01/86 00:00","12/31/94 00:00",50,250,"SO4",1)
maxd2  <- maxDistSites("01/01/07 00:00","12/31/14 00:00",50,230,"SO4",1)
maxd3  <- maxDistSites("01/01/07 00:00","12/31/14 00:00",50,230,"NO3",1)
maxd4  <- maxDistSites("01/01/07 00:00","12/31/14 00:00",50,230,"Na",1)
maxd5  <- maxDistSites("01/01/07 00:00","12/31/14 00:00",50,230,"NH4",1)
```

Acknowledgments

J. Lee was funded by the Air Force Office of Scientific Research (Complex Networks program), FA9550-19-1-0175. H. Al-Thani was funded by the Qatar National Research Fund (Graduate Sponsorship Research Program). The authors are very grateful to Dr. Martin Shafer and Robert Larson for helping us gain access to the NADP/NTN data in a convenient form.

5 Appendix

Appendix A

United States
WASHINGTON
MONTANA
NORTH DAKOTA
MINNESOTA
SOUTH DAKOTA
WISCONSIN
OREGON
IDAHO
WYOMING
MICHIGAN
Ottawa
Montreal
Toronto
MAINE
NEW YORK
Chicago
IOWA
NEBRASKA
ILLINOIS
INDIANA
OHIO
PENN
Philadelphia
NEVADA
UTAH
COLORADO
KANSAS
MISSOURI
San Francisco
CALIFORNIA
Las Vegas
KENTUCKY
VIRGINIA
WEST VIRGINIA
TENNESSEE
NORTH CAROLINA
Los Angeles
San Diego
ARIZONA
NEW MEXICO
OKLAHOMA
ARKANSAS
Dallas
MISSISSIPPI
ALABAMA
GEORGIA
SOUTH CAROLINA
TEXAS
LOUISIANA
Houston
FLORIDA
Gulf of Mexico
Gulf of California

References

[AFLW99] Kurt M. Anstreicher, Marcia Fampa, Jon Lee, and Joy Williams. Using continuous nonlinear relaxations to solve constrained maximum-entropy sampling problems. *Mathematical Programming*, 85(2, Ser. A):221–240, 1999.

[AL04] Kurt M. Anstreicher and Jon Lee. A masked spectral bound for maximum-entropy sampling. In *mODa 7—Advances in model-oriented design and analysis*, Contrib. Statist., pages 1–12. Physica, Heidelberg, 2004.

[Ans18a] Kurt M. Anstreicher. Efficient solution of maximum-entropy sampling problems. Preprint available at: `https://www.biz.uiowa.edu/faculty/anstreicher/papers/linx.pdf`, July 2018.

[Ans18b] Kurt M. Anstreicher. Maximum-entropy sampling and the boolean quadric polytope. *Journal of Global Optimization*, 72(4):603–618, 2018.

[BL07] Samuel Burer and Jon Lee. Solving maximum-entropy sampling problems using factored masks. *Mathematical Programming*, 109(2-3, Ser. B):263–281, 2007.

[BLZ94] Philip J. Brown, Nhu D. Le, and James V. Zidek. Multivariate spatial interpolation and exposure to air pollutants. *Canad. J. Statist.*, 22(4):489–509, 1994.

[FL00] Valerii Fedorov and Jon Lee. Design of experiments in statistics. In *Handbook of semidefinite programming*, volume 27 of *Internat. Ser. Oper. Res. Management Sci.*, pages 511–532. Kluwer Acad. Publ., Boston, MA, 2000.

[GLSZ93] Peter Guttorp, Nhu D. Le, Paul D. Sampson, and James V. Zidek. Using entropy in the redesign of an environmental monitoring network. In *Multivariate environmental statistics*, volume 6 of *North-Holland Ser. Statist. Probab.*, pages 175–202. North-Holland, Amsterdam, 1993.

[Goe11] Georg M. Goerg. Lambert w random variables - a new family of generalized skewed distributions with applications to risk estimation. *Annals of Applied Statistics*, 5(3):2197–2230, 2011.

[Goe16] Georg M. Goerg. *LambertW: Probabilistic Models to Analyze and Gaussianize Heavy-Tailed, Skewed Data*, 2016. R package version 0.6.4.

[HLW01] Alan J. Hoffman, Jon Lee, and Joy Williams. New upper bounds for maximum-entropy sampling. In *mODa 6—advances in model-oriented design and analysis (Puchberg/Schneeberg, 2001)*, Contrib. Statist., pages 143–153. Physica, Heidelberg, 2001.

[KGZ14] Selcuk Korkmaz, Dincer Goksuluk, and Gokmen Zararsiz. Mvn: An r package for assessing multivariate normality. *The R Journal*, 6(2):151–162, 2014.

[KLQ95] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

[Lee98] Jon Lee. Constrained maximum-entropy sampling. *Operations Research*, 46(5):655–664, 1998.

[Lee12] Jon Lee. *Encyclopedia of Environmetrics, A.H. El-Shaarawi and W.W. Piegorsch, eds.*, chapter Maximum entropy sampling, 2nd edition, pages 1570–1574. Wiley, 2012.

[LW03] Jon Lee and Joy Williams. A linear integer programming bound for maximum-entropy sampling. *Mathematical Programming*, 94(2-3, Ser. B):247–256, 2003.

[NAD18] NADP. National Acidic Deposition Program, National Trends Network. `https://nadp.slh.wisc.edu/ntn/`, 2018.

[SW87] Michael C. Shewry and Henry P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

[SW00] Paola Sebastiani and Henry P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157, 2000.

[ZSL00] James V. Zidek, Weimin Sun, and Nhu D. Le. Designing and integrating composite networks for monitoring multivariate Gaussian pollution fields. *J. Roy. Statist. Soc. Ser. C*, 49(1):63–79, 2000.