1. Abstract
2. Introduction

The country of Bangladesh lies on the Ganges-Brahmaputra (GB) deltaplain beneath the shadow of the Himalayas. The country is one of the most densely populated countries in the world with 156.6 million people living within an area of 147,500 km2 (approximately the size of Iowa). In the 1970s, surface water was widely abandoned for groundwater due to surface water contamination by pathogens and industrial and organic pollutants. The consequences of this switch were not discovered until the mid-1990s. Nearly one-third of shallow wells in Bangladesh are contaminated with naturally occurring arsenic. These wells exceed the Bangladesh drinking water standard of 50 µg l-1 and almost half exceed the World Health Organization standard of 10 µg l-1 (BGS 2001; Kinniburgh and Kosmus, 2002; WHO, 2011). Sustained low-doses of arsenic above these thresholds can lead to lead to arsenicosis and a variety of related cancers (WHO, 2011). However, there are no good alternatives to drinking water. There is a need to be able to predict arsenic contamination more effectively. Successful prediction will allow better resource allocation to help solve the widespread problem of arsenic contamination.

*The Setting*

These two rivers dissect separate terrains of the Himalayas. With the erosion of their respective basins, the rivers bring sediment sourced from their respective terrains.

1. Literature Review

There has been much research into the geochemical modeling of arsenic mobility within the delta (Harvey et al., 2002; Ravenscroft et al., 2005; Fendorf et al., 2010) which has elucidated the larger scale spatial patterns of arsenic contamination. However, on a local scale, it is still quite difficult to predict arsenic occurrence. Anecdotally, wells separated by as little as 10 m at a similar depth can have vastly different arsenic concentrations. Some research (Gelman et al., 2004; Winkel et al., 2008) have attempted to predict groundwater arsenic contamination, but most are either at a very local or regional scale.

1. Methods

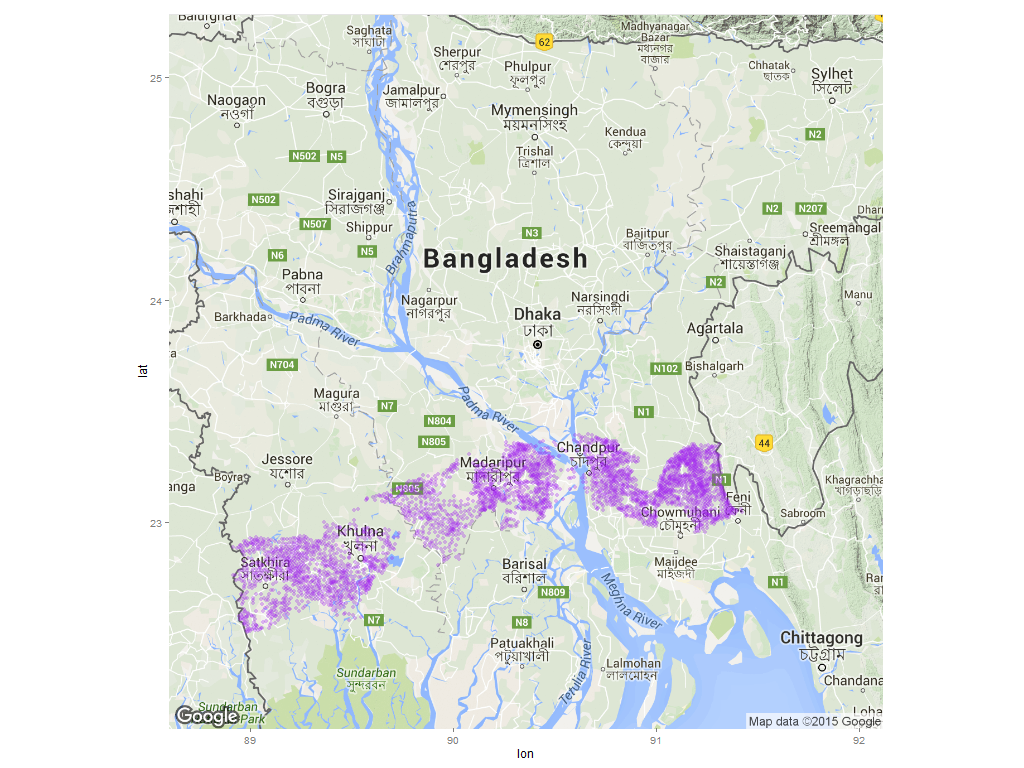
Here, we use the Bangladesh Arsenic Mitigation and Water Supply Program (BAMSWP) dataset (n=4.69x106) of field kit measurements of arsenic concentrations in groundwater. We attempt to build multiple types of regression models to predict arsenic content based on three predictors – longitude, latitude, and depth of the well.

*Bangladesh Arsenic Mitigation and Water Supply Program Dataset*

The BAMWSP dataset contains arsenic concentration for ~5 million wells throughout the country of Bangladesh. The data includes the well depth (ft), arsenic concentration (ppb), and geocode. The geocodes are a derived unit to describe nested administrative units in Bangladesh. These geocodes are associated with the mouza or township unit. In order to assign a more definitive x and y location for wells, we assign the longitude and latitude of the mouza centroid to the wells within each mouza. This results in many wells having the same longitude and latitude. We then filter errant data points (i.e. unrealistic values for predictors) and NAs by removing the entire observation. We also add new response variables derived from arsenic concentration. The final dataset contains arsenic concentration (ppb) as both continuous (As and log(As)) and categorical (binary variables where As >10 ppb and As >50ppb), well depth (ft), longitude, and latitude (Table 1).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Response Variables | | | | Predictor Variables | | |
| As (ppb) | log(As) | As > 10 ppb | As > 50 ppb | Depth (ft) | Longitude | Latitude |
| 25 | 3.2188 | 1 | 0 | 180 | 90.44368 | 23.12708 |
| 51 | 3.9318 | 1 | 1 | 180 | 90.44368 | 23.12708 |
| 0 | -6.907755 | 0 | 0 | 800 | 90.44368 | 23.12708 |

**Table 1 |** Sample of arsenic dataset. The dataset was derived from the BAMWSP data.

For the purpose of this study, we chose a subset (n=4.66x105) of the original data (Fig. 1) for two reasons: 1) it was computationally less expensive to implement and test models and 2) the subset represents a portion of the country where we have collected stratigraphy data which we will use as a predictor in future model iterations.

*Predictor Selection*

Predictors were selected using best subset selection as there were only three total predictors. All predictors were selected regardless of the metric used to evaluated the best model (Fig. 2).

*Regression Modeling of Arsenic Concentrations*

We modeled arsenic concentrations using linear regression, regression trees, and non-linear polynomial regressions. We also applied log transformations to the response variable in some of our models.

*Regression Modeling of Arsenic Concentrations*

We modeled arsenic concentrations using multiple regression with predictors of depth, latitude, and longitude.

|  |  |
| --- | --- |
| Null1 | 178.9731 |
| Null 2 | 217.3598 |
| Linear 1 | 175.3610 |
| Linear 2 | 213.5510 |
| Rtree 1 | 140.0899 |
| Rtree 2 | 180.2832 |
| Poly 1 | 175.3411 |
| Poly 2 | 175.1521 |
| Poly 3 | 175.3377 |
| Poly 4 | 163.7737 |

1. Discussion
2. References
3. Appendix

### LOAD LIBRARIES ###

library**(**pacman**)**

pacman**::**p\_load**(**ggplot2,dplyr,rpart**)**

library**(**e1071**)**

library**(**leaps**)**

### LOAD DATA ###

setwd**(**'C:/Projects/Vanderbilt/bng\_arsenic/'**)**

as.df **=** tbl\_df**(**read.csv**(**'data/bamwsp.csv',header**=**T**))**

as.df **=** rename**(**as.df,As\_ppb **=** arsenic\_ppb**)**

xy.df **=** tbl\_df**(**read.csv**(**'data/Mouza\_25km.csv'**))**

xy.df **=** rename**(**xy.df,geocode **=** GEO2,lon **=** X\_COORD,lat **=** Y\_COORD**)**

df **=** left\_join**(**as.df,xy.df,by**=**'geocode'**)**

df**$**geocode **=** as.factor**(**as.df**$**geocode**)**

d **=** df %>%

filter**(!**is.na**(**lon**))** %>%

filter**(!**is.na**(**lat**))** %>%

filter**(**depth\_ft**>**0**)** %>%

filter**(**depth\_ft**<**3000**)** %>%

filter**(**As\_ppb**<**1000**)** %>%

select**(**As\_ppb,depth**=**depth\_ft,lon,lat,geocode**)** %>%

mutate**(**As\_10 **=** as.factor**(**ifelse**(**As\_ppb**>=**10,1,0**)))** %>%

mutate**(**As\_50 **=** as.factor**(**ifelse**(**As\_ppb**>=**50,1,0**)))** %>%

mutate**(**as **=** ifelse**(**As\_ppb**==**0,0.001,As\_ppb**)**,log\_as**=**log**(**as**))**

### VISUALIZE DATA ###

# Pairs plot with 10% of data

d.sample **=** sample\_frac**(**d,0.1,replace**=**F**)**

pairs**(~**as**+**log\_as**+**lon**+**lat,data**=**d.sample,upper.panel**=NULL**,pch**=**'.'**)**

# Histogram of >0 As Concentrations

d.gt0As **=** filter**(**d,as**>**0**)**

hist**(**d.gt0As**$**as,breaks**=**60,xlab**=**'As Concentration (ppb)',ylab**=**'counts',col**=**2,main**=**'Histogram of Non-Zero As Concentrations'**)**

## Plot As Concentrations vs Other Variables

# Plot As concentrations vs Depth

plot**(**d**$**as,**-**d**$**depth,axes**=**F,ann**=**F,col**=**'darkgrey'**)**

axis**(**3**)**

axis**(**2**)**

box**()**

mtext**(**'Arsenic Concentration (ppb)',side**=**3,line**=**3,cex **=** par**(**"cex.lab"**))**

mtext**(**'Depth (ft)',side**=**2,line**=**3,cex **=** par**(**"cex.lab"**))**

# Plot As concentrations vs Lon

plot**(**d**$**lon,d**$**as,xlab **=** 'Longitude',ylab**=**'Arsenic Concentration (ppb)',col**=**'darkgrey'**)**

# Plot As concentrations vs Lat

plot**(**d**$**as,d**$**lat,ylab **=** 'Latitude',xlab**=**'Arsenic Concentration (ppb)',col**=**'darkgrey'**)**

## Determine variables using Best Subset Selection

regfit.full **=** regsubsets**(**as**~**depth**+**lon**+**lat,d**)**

reg.summary **=** summary**(**regfit.full**)**

# Plot model statistics

par**(**mfrow**=**c**(**2,2**))**

plot**(**reg.summary**$**rss ,xlab**=**"Number of Variables",ylab**=**"RSS",type**=**"l"**)**

best.rss **=** which.min**(**reg.summary**$**rss**)**

points**(**best.rss,reg.summary**$**rss**[**best.rss**]**,col**=**"red",cex**=**2,pch**=**20**)**

plot**(**reg.summary**$**adjr2 ,xlab **=**" Number of Variables ",ylab**=**" Adjusted RSq",type**=**"l"**)**

best.r2 **=** which.max**(**reg.summary**$**adjr2**)**

points**(**best.r2,reg.summary**$**adjr2**[**best.r2**]**,col**=**"red",cex**=**2,pch**=**20**)**

plot**(**reg.summary**$**cp ,xlab **=**"Number of Variables",ylab**=**"Cp",type**=**'l'**)**

best.cp**=**which.min**(**reg.summary**$**cp**)**

points **(**best.cp,reg.summary**$**cp**[**best.cp**]**,col**=**"red",cex**=**2,pch**=**20**)**

plot**(**reg.summary**$**bic,xlab **=**"Number of Variables",ylab**=**"BIC",type**=**'l'**)**

best.bic **=** which.min**(**reg.summary**$**bic**)**

points**(**best.bic,reg.summary**$**bic**[**best.bic**]**,col**=**"red",cex**=**2,pch**=**20**)**

par**(**mfrow**=**c**(**2,2**))**

plot**(**regfit.full,scale **=**"r2"**)**

plot**(**regfit.full,scale **=**"adjr2"**)**

plot**(**regfit.full,scale **=**"Cp"**)**

plot**(**regfit.full,scale **=**"bic"**)**

### MODELS ###

# num of folds

k **<-** 10

# create index

folds **<-** rep\_len**(**1**:**k, nrow**(**d**))**

# shuffle index using sample

folds **<-** sample**(**folds, nrow**(**d**))**

# number of models

nmodels **<-** 11

model.error **<-** matrix**(**data**=NA**, nrow**=**nrow**(**d**)**, ncol**=**nmodels**)**

**for** **(**i **in** 1**:**k**){**

# select rows in fold

fold **<-** which**(**folds**==**i**)**

# subset training data

train **<-** d**[-**fold,**]**

# model 1 = null model of As

null1 **<-** mean**(**train**$**as**)**

# model 2 = null model of log(As)

null2 **<-** mean**(**train**$**log\_as**)**

# model 3 = linear model of As

linear1 **<-** lm**(**as **~** depth **+** lat **+** lon, data**=**train**)**

# model 4 = linear model of log(As)

linear2 **<-** lm**(**log\_as **~** depth **+** lat **+** lon, data**=**train**)**

# model 5 = regression tree of As

rtree1 **<-** rpart**(**as **~** depth **+** lat **+** lon, data**=**train**)**

# model 6 = regression tree of log(As)

rtree2 **<-** rpart**(**log\_as **~** depth **+** lat **+** lon, data**=**train**)**

# model 7 = polynomial model

poly1 **<-** lm**(**as **~** poly**(**depth,3**)** **+** lat **+** lon, data**=**train**)**

# model 8 = polynomial model

poly2 **<-** lm**(**as **~** poly**(**depth,4**)** **+** lat **+** lon, data**=**train**)**

# model 9 = polynomial model

poly3 **<-** lm**(**as **~** depth **+** poly**(**lat,2**)** **+** lon, data**=**train**)**

# model 10 = SVM

svm **<-** svm**(**as **~** depth **+** lat **+** lon, data**=**train**)**

# test set

test **<-** d**[**fold,**]**

# store observed and predictions for folds

model.error**[**fold,1**]** **=** exp**(**test**$**log\_as**)**

model.error**[**fold,2**]** **=** null1

model.error**[**fold,3**]** **=** exp**(**null2**)**

model.error**[**fold,4**]** **=** predict**(**linear1,test**)**

model.error**[**fold,5**]** **=** exp**(**predict**(**linear2,test**))**

model.error**[**fold,6**]** **=** predict**(**rtree1,test**)**

model.error**[**fold,7**]** **=** exp**(**predict**(**rtree2,test**))**

model.error**[**fold,8**]** **=** predict**(**poly1,test**)**

model.error**[**fold,9**]** **=** predict**(**poly2,test**)**

model.error**[**fold,10**]** **=** predict**(**poly3,test**)**

model.error**[**fold,11**]** **=** predict**(**svm,test**)**

**}**

# store errors in a dataframe

model.results **<-** data.frame**(**obs**=**model.error**[**,1**]**,

null1**=**model.error**[**,2**]**,

null2**=**model.error**[**,3**]**,

linear1**=**model.error**[**,4**]**,

linear2**=**model.error**[**,5**]**,

rtree1**=**model.error**[**,6**]**,

rtree2**=**model.error**[**,7**]**,

poly1**=**model.error**[**,8**]**,

poly2**=**model.error**[**,9**]**,

poly3**=**model.error**[**,10**]**,

svm**=**model.error**[**,11**])**

# rmse function

rmse **<-** **function(**obs,est**){**sqrt**(**mean**((**obs**-**est**)^**2,na.rm**=**T**))}**

# calculate rmse for models

rmse.all **<-** apply**(**model.results**[**,2**:**nmodels**]**,2, rmse, obs**=**model.results**[**,1**])**

### PLOTS ###

# Models with RMSE

plot**(**rmse.all,xaxt**=**'n',xlab**=**'Model',ylab**=**'RMSE',ylim**=**range**(**0**:**max**(**rmse.all**)))**

axis**(**1,at**=**1**:**10,labels**=**names**(**rmse.all**))**

points**(**5,rmse.all**[**5**]**,col**=**"red",cex**=**2,pch**=**20**)**

# Best Model

plot**(**rtree1, uniform**=TRUE**, main**=**"Regression Tree for Arsenic Concentration"**)**

text**(**rtree1, all**=TRUE**, cex**=**.8**)**