MAE vs Confidence

Karsten Maurer

February 7, 2019

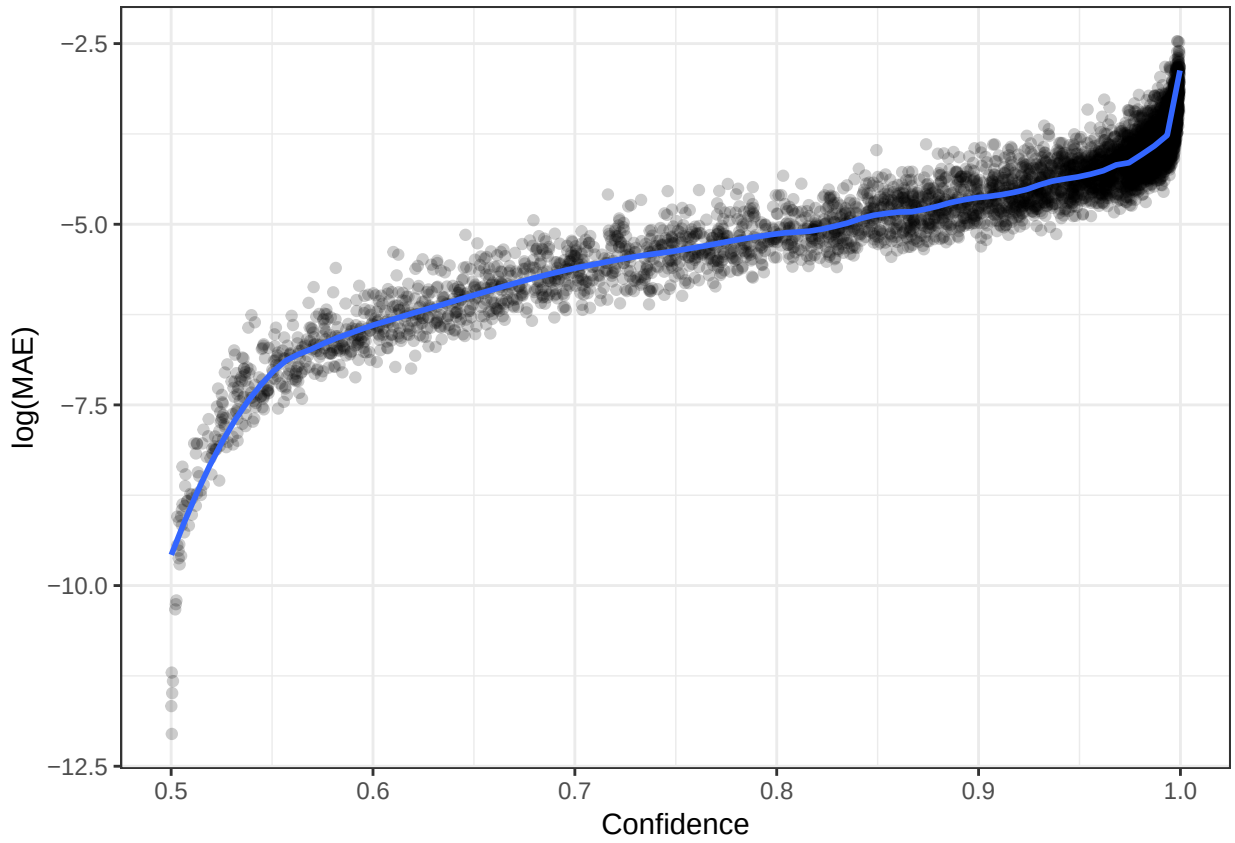
```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

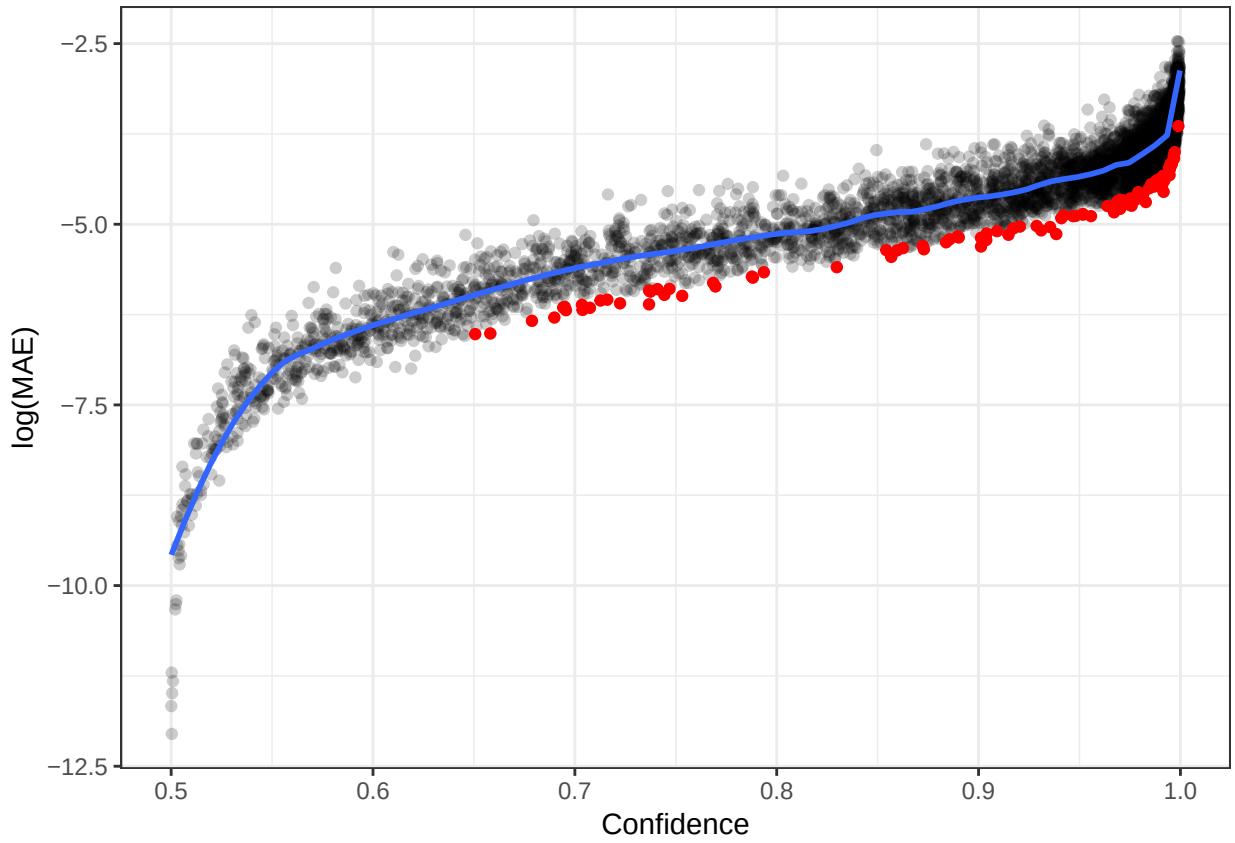
```
setwd("C:\\Users\\maurerkt\\Documents\\GitHub\\uuutils\\adversarialImages")
dat <- read.csv("dataForKarsten.csv")
head(dat)
```

```
##   X Prediction Confidence True.Label        X1         X2         X3
## 1 0          1  0.5866472          0 -26.33029  -9.684671 -10.217214
## 2 1          1  0.7709508          0 -52.80538 -34.758061   1.397407
## 3 2          1  0.6592368          0 -49.67168 -27.209196  12.035836
## 4 3          1  0.9970490          1  59.77651  12.934427   2.282467
## 5 4          1  0.9219793          1 -54.24267  34.713743 -17.669494
## 6 5          1  0.5316441          0 -27.88494 -15.428062 -20.249635
##            X4          X5 Misclassified         MAE
## 1  -2.2273409  -0.2074427          True 0.001096648
## 2  11.3922865 -13.4802990          True 0.004058172
## 3  -0.5962296  -4.6825862          True 0.002275917
## 4 -13.4526526  28.7890923         False 0.024314610
## 5 -12.4584927  11.6628733         False 0.008625720
## 6  -7.3316953   7.8422969          True 0.001108090
```

```
# # plot (no log)
# ggplot(aes(x=Confidence, y=MAE), data=dat)+
#   geom_point(alpha=.2)+
#   stat_smooth(method="loess", span = 0.1, se=FALSE)+
#   theme_bw()

# plot (with log)
ggplot(aes(x=Confidence, y=log(MAE)), data=dat)+
  geom_point(alpha=.2)+
  stat_smooth(method="loess", span = 0.1, se=FALSE)+
  theme_bw()
```




```
# create log MAE column
dat$logMAE <- log(dat$MAE)
# fit loess smoother - a non-parametric "sliding average" line, then recode residuals of all points
line <- loess(logMAE ~ Confidence, data=dat, span=.1)
dat$resids <- line$residuals
# Alternatively could calculate from plugging data back through prediction function
dat$resids2 <- dat$logMAE - predict(line, newdata=dat)
#!# Note: any fitted model that allows predictions for logMAE could replace loess

# Take top B largest negative resids with conf > .65
B=100
queryset <- dat %>%
  dplyr::filter(Confidence > .65) %>%
  arrange(resids) %>%
  head(100)

# which images are picked?
ggplot()+
  geom_point(aes(x=Confidence, y=log(MAE)), data=dat,
             alpha=.2)+
  stat_smooth(aes(x=Confidence, y=log(MAE)), data=dat,
              method="loess", span = 0.1, se=FALSE)+
  geom_point(aes(x=Confidence, y=log(MAE)), data=queryset,
             color="red")+
  theme_bw()
```




```
# what is the SDR?
sdr = sum(queryset$Misclassified == "True")/(B-sum(queryset$Confidence))
sdr
```

```
## [1] 3.685874
```