GlobalAnalysis

Kevin Ros

2/16/2020

Loading the original and synthetic data from the previous week.

```
load("synthetic_one.Rda")
```

Evaluating the propensity score measure:

Merging the two datasets, and adding T flag (0 = original, 1 = synthetic)

```
zero_col = rep(0,nrow(synthetic_one))
tmp_orig = data.frame(synthetic_one$OriginalIncome)
tmp_orig$T = zero_col
colnames(tmp_orig)[colnames(tmp_orig) == "synthetic_one.OriginalIncome"] <- "Income"
one_col = rep(1,nrow(synthetic_one))
tmp_syn = data.frame(synthetic_one$logIncome_syn)
tmp_syn$T = one_col
colnames(tmp_syn)[colnames(tmp_syn) == "synthetic_one.logIncome_syn"] <- "Income"
merged_data = rbind(tmp_orig, tmp_syn)
```

Fitting the logistic regression model

```
# T is predicted by income
log_reg_model = glm(T ~ Income , data = merged_data, family = "binomial")

# C = ratio of synthetic data to all data
N = nrow(merged_data)
c = 1/2

# Calculate predictions for all Income values (original and synthetic)
pred = data.frame(merged_data$Income)
pred$rankP = predict(log_reg_model, newdata = pred$Income, type = "response")

# Calculate the propensity score measure
# Note that I am not sure what data to use / what "the estimated betas, which are obtained by maximum l
# so I fed the training data back into the model and used the resulting estimations as the pis
p_score = 0
for (row in 1:nrow(pred)) {
  p_score = p_score + (pred[row, "rankP"] - c)^2
}
p_score = p_score * (1/N)
p_score
```

```
## [1] 0.1168257
```

Overall, I think this is a reasonable result. About halfway between 0 (simulated identical to original) and 0.25 (simulated completely distinguishable from original). But again, I am not sure that I correctly calculated this score.

Evaluating the cluster analysis measure:

```
# Number of groups is a guess; c is ratio of simulated data to total data
num_groups = 10
c = 1/2
```

Fitting the kmeans cluster model to the data

```
fit = kmeans(merged_data$Income,num_groups)
aggregate(merged_data$Income,by=list(fit$cluster),FUN=mean)
```

```
##    Group.1         x
## 1        1 11.001798
## 2        2  6.944517
## 3        3  9.193206
## 4        4 11.924290
## 5        5  9.837816
## 6        6 10.459439
## 7        7  8.546776
## 8        8 11.469270
## 9        9 12.676973
## 10      10  7.824806
```

```
merged_data <- data.frame(merged_data, fit$cluster)
```

Calculating the score for each cluster, and aggregating the results

```
cluster_score = 0
for (group in 1:num_groups) {
  # Wasn't sure how to find the weight of each cluster, but I imagine its somehow found through the fit
  group_data_orig = merged_data[merged_data$fit.cluster == group & merged_data$T == 0,]
  group_data_syn = merged_data[merged_data$fit.cluster == group & merged_data$T == 1,]
  cluster_score = cluster_score + ((nrow(group_data_orig) / (nrow(group_data_orig) + nrow(group_data_sy
}
cluster_score =  cluster_score * 1/num_groups
cluster_score
```

```
## [1] 0.1512733
```

I'm not sure what a reasonable score is, but I think it has the same properties as the propensity score.

Evaluating emperical CDF measures:

```
# I have no clue where to start.
```