Final Project Rough Draft

Daniel Lee

April 3, 2017

1 Dagitty

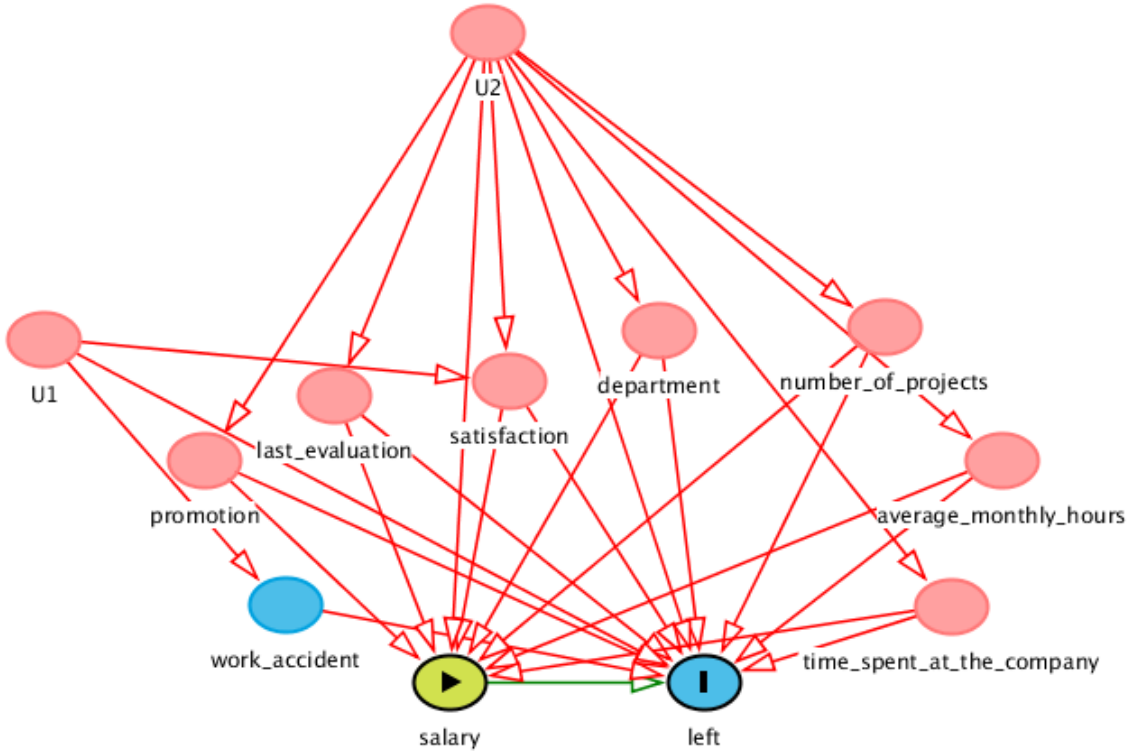

Figure 1: Some independence assumptions made

I'm looking at the causal relationship between salary and whether or not someone leaves the company. I need to think more about the causal model. Appropriate assumptions have not been made for identifiability.

Causal parameter $\Psi(P_{U,X})$ is the average treatment effect $E_{U,X}(Y_1 - Y_0)$. Under the assumption that W fulfills the backdoor criteria and the positivity assumption, we have the following:

$$\begin{aligned}\Psi^F(P_{U,X}) &= \Psi(P_0) \\ &= \Psi(Q_0) \\ &= \sum_w E_0[E_0(Y|A=1,W=w) - E_0(Y|A=0,W=w)] \\ &= \sum_w [E_0(Y|A=1,W=w) - E_0(Y|A=0,W=w)]P_0(W=w)\end{aligned}$$

I will estimate $\Psi(Q_0)$ using the substitution estimator:

$$\begin{aligned}\hat{\Psi}(P_n) &= \Psi(Q_n) \\ &= \frac{1}{n}\sum_{i=1}^{n}[\bar{Q}_n(1,W_i) - \bar{Q}_n(0,W_i)]\end{aligned}$$

where $\bar{Q}_n(A,W)$ is an estimator of $E_0(Y|A,W)$.

$\bar{Q}_n(A,W)$ is obtained using super learner. Based on the SuperLearner, from random forest, stepwise selection, stepwise selection based on AIC, glmnet, step forward regression, and glm, random forest had the lowest MSE. SuperLearner chooses random forest with weight = 1. Results are attached below.

Final Project

Stat 135

April 3, 2017

```
library(SuperLearner)
```

```
## Loading required package: nnls
```

```
## Super Learner
```

```
## Version: 2.0-21
```

```
## Package created on 2016-11-11
```

```
data <- read.table("HR_comma_sep.csv", sep = ",", header = TRUE)
head(data)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                  157
## 2               0.80            0.86              5                  262
## 3               0.11            0.88              7                  272
## 4               0.72            0.87              5                  223
## 5               0.37            0.52              2                  159
## 6               0.41            0.50              2                  153
##   time_spend_company Work_accident left promotion_last_5years sales salary
## 1                  3             0    1                     0 sales    low
## 2                  6             0    1                     0 sales medium
## 3                  4             0    1                     0 sales medium
## 4                  5             0    1                     0 sales    low
## 5                  3             0    1                     0 sales    low
## 6                  3             0    1                     0 sales    low
```

```
names(data)
```

```
##  [1] "satisfaction_level"    "last_evaluation"
##  [3] "number_project"        "average_montly_hours"
##  [5] "time_spend_company"    "Work_accident"
##  [7] "left"                  "promotion_last_5years"
##  [9] "sales"                 "salary"
```

```
n <- nrow(data)
n
```

```
## [1] 14999
```

```
# transform sales to numeric
sales <- factor(data$sales)
nlevels(sales)
```

```
## [1] 10
```

```
sales <- as.numeric(factor(sales, labels = 1:nlevels(sales)))
summary(sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   8.000   6.936   9.000  10.000
```

```r
# transform salary to numeric
salary <- factor(data$salary)
nlevels(salary)
```

```
## [1] 3
```

```r
salary <- as.numeric(factor(salary, labels = 1:nlevels(salary)))
summary(salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.347   3.000   3.000
```

```r
# transform data to contain sales and salary as numeric
data <- subset(data, select = c(-sales, -salary))
head(data)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                   157
## 2               0.80            0.86              5                   262
## 3               0.11            0.88              7                   272
## 4               0.72            0.87              5                   223
## 5               0.37            0.52              2                   159
## 6               0.41            0.50              2                   153
##   time_spend_company Work_accident left promotion_last_5years
## 1                  3             0    1                     0
## 2                  6             0    1                     0
## 3                  4             0    1                     0
## 4                  5             0    1                     0
## 5                  3             0    1                     0
## 6                  3             0    1                     0
```

```r
data <- data.frame(data, sales = sales, salary = salary)
summary(data)
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##  time_spend_company Work_accident         left
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 3.000     Median :0.0000   Median :0.0000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000
##  promotion_last_5years     sales            salary
##  Min.   :0.00000       Min.   : 1.000   Min.   :1.000
##  1st Qu.:0.00000       1st Qu.: 5.000   1st Qu.:2.000
##  Median :0.00000       Median : 8.000   Median :2.000
##  Mean   :0.02127       Mean   : 6.936   Mean   :2.347
##  3rd Qu.:0.00000       3rd Qu.: 9.000   3rd Qu.:3.000
##  Max.   :1.00000       Max.   :10.000   Max.   :3.000
```

```r
SL.library<- c("SL.randomForest",
               "SL.step",
               "SL.stepAIC",
               "SL.glmnet",
               "SL.step.forward",
               "SL.glm")

# remove the outcome variable "left"
X <- subset(data, select = -left)

# run superlearner
SL.out <- SuperLearner(Y = data$left, X = X, SL.library = SL.library, family = "binomial",
                       cvControl=list(V=10))
```

```
## Loading required package: glmnet
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-5
## Loading required package: randomForest
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
## Loading required package: MASS
```

```r
# evaluate superlearner
CV.SL.out<- CV.SuperLearner(Y = data$left, X = X, SL.library=SL.library, family='binomial',
                            innerCvControl=list(V=10), cvControl=list(V=10))
```

```
## Warning in CV.SuperLearner(Y = data$left, X = X, SL.library = SL.library, :
## Only a single innerCvControl is given, will be replicated across all cross-
## validation split calls to SuperLearner
```

```r
summary(CV.SL.out)
```

```
##
## Call:
## CV.SuperLearner(Y = data$left, X = X, family = "binomial", SL.library = SL.library,
##     cvControl = list(V = 10), innerCvControl = list(V = 10))
##
## Risk is based on: Mean Squared Error
##
## All risk estimates are based on V =  10
##
##            Algorithm       Ave        se       Min      Max
##        Super Learner 0.0086013 0.0005931 0.0059572 0.013669
##          Discrete SL 0.0086013 0.0005931 0.0059572 0.013669
##  SL.randomForest_All 0.0086013 0.0005931 0.0059572 0.013669
##          SL.step_All 0.1454877 0.0017114 0.1370137 0.154356
##       SL.stepAIC_All 0.1455392 0.0017112 0.1372322 0.154475
##        SL.glmnet_All 0.1454361 0.0017067 0.1370426 0.154138
##  SL.step.forward_All 0.1454877 0.0017114 0.1370137 0.154356
##           SL.glm_All 0.1455014 0.0017114 0.1370346 0.154364
```

```
CV.SL.out$AllSL
```

```
## $`1`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.009756537    1
## SL.step_All         0.144731690    0
## SL.stepAIC_All      0.144773323    0
## SL.glmnet_All       0.144680357    0
## SL.step.forward_All 0.144731690    0
## SL.glm_All          0.144738058    0
##
## $`2`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.009438799    1
## SL.step_All         0.145418210    0
## SL.stepAIC_All      0.145452811    0
## SL.glmnet_All       0.145367972    0
## SL.step.forward_All 0.145418210    0
## SL.glm_All          0.145453805    0
##
## $`3`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.009977372    1
## SL.step_All         0.145948933    0
## SL.stepAIC_All      0.146037883    0
## SL.glmnet_All       0.145908422    0
## SL.step.forward_All 0.145948933    0
## SL.glm_All          0.145984149    0
##
## $`4`
```

```
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                           Risk Coef
## SL.randomForest_All 0.00998544    1
## SL.step_All         0.14471272    0
## SL.stepAIC_All      0.14474613    0
## SL.glmnet_All       0.14465167    0
## SL.step.forward_All 0.14471272    0
## SL.glm_All          0.14472944    0
##
## $`5`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.008815925    1
## SL.step_All         0.145393430    0
## SL.stepAIC_All      0.145542601    0
## SL.glmnet_All       0.145352890    0
## SL.step.forward_All 0.145393430    0
## SL.glm_All          0.145426079    0
##
## $`6`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.009940334    1
## SL.step_All         0.146201075    0
## SL.stepAIC_All      0.146189514    0
## SL.glmnet_All       0.146157477    0
## SL.step.forward_All 0.146201075    0
## SL.glm_All          0.146227523    0
##
## $`7`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
```

```
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.009963532    1
## SL.step_All         0.145650573    0
## SL.stepAIC_All      0.145734663    0
## SL.glmnet_All       0.145619360    0
## SL.step.forward_All 0.145650573    0
## SL.glm_All          0.145701667    0
##
## $`8`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.009642579    1
## SL.step_All         0.146273499    0
## SL.stepAIC_All      0.146236720    0
## SL.glmnet_All       0.146210991    0
## SL.step.forward_All 0.146273499    0
## SL.glm_All          0.146273162    0
##
## $`9`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
##
##
##                            Risk Coef
## SL.randomForest_All 0.009514821    1
## SL.step_All         0.145351674    0
## SL.stepAIC_All      0.145463418    0
## SL.glmnet_All       0.145300072    0
## SL.step.forward_All 0.145351674    0
## SL.glm_All          0.145371875    0
##
## $`10`
##
## Call:
## SuperLearner(Y = cvOutcome, X = cvLearn, newX = cvValid, family = family,
##     SL.library = SL.library, method = method, id = cvId, verbose = verbose,
##     control = control, cvControl = valid[[2]], obsWeights = cvObsWeights,
##     env = env)
```

```
##
##
##                            Risk Coef
## SL.randomForest_All 0.01025245    1
## SL.step_All         0.14532703    0
## SL.stepAIC_All      0.14538654    0
## SL.glmnet_All       0.14527811    0
## SL.step.forward_All 0.14532703    0
## SL.glm_All          0.14536054    0
```