ORFE 525: Statistical Learning and Nonparametric Estimation
Homework 2

Zachary Hervieux-Moore

Thursday 23rd March, 2017

Exercise 1: This is an effort to reproduce the Moneyball analysis using R.

a) Fit salary by variables on the right of `POS` using tree regression via `rpart()`. Compare the mean square errors on the training and testing data separately for the two methods: tree regression with or without pruning. You can choose the parameters of the methods yourself. Plot your fitted trees via `prp()`.

b) Fit salary by variables on the right of `POS` using tree regression via `randomForest()` with different number of baggings B. Plot the mean square errors on the training and testing data separately with B as the x-axis. You canc hoose the other parameters of the methods yourself.

c) Compare the above two results and which model performs better (in terms of testing error)? Using the better performed model to find the player most undervalued (i.e., the predicted salary has the largest positive gap from the real salary). Check his salary in the following years in *Baseball Prospectus*. As a general manager, what do you think is the possible reasons that he is undervalued.

Answer: The code used for this question is appended at the end.

a) The three figures show the tree generated for 3 possible pruning values. Figure 1 is no pruning, Figure 2 is cp = 0.1, and Figure 3 is cp = 0.3. Notice that even with cp = 0.3, the tree reduces down to a single node. The mean square errors are in Table 1.

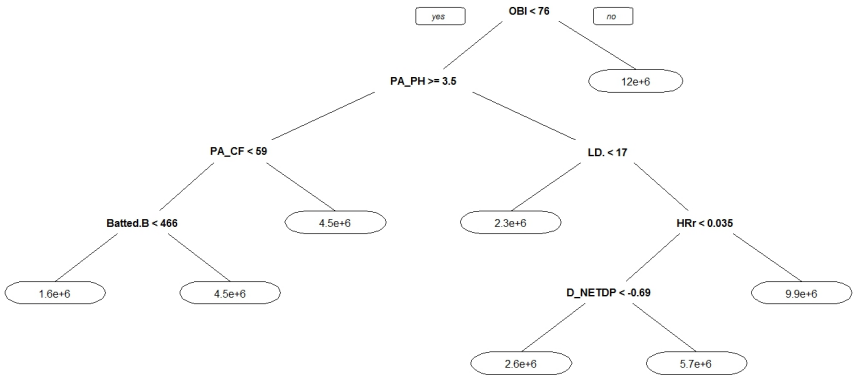

Figure 1: Tree with no Pruning

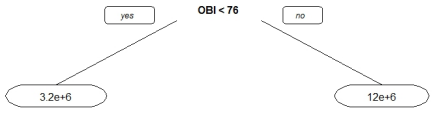

Figure 2: Tree with cp = 0.1

Figure 3: Tree with cp = 0.3

Table 1: MSE Values

	No pruning	cp = 0.1	cp = 0.3
Training Set	36198241	44735393	52769876
Test Set	72784943	74445712	73588157

b) Figure 4 shows the MSE values as you vary B from 10 to 200 in increments of 5.

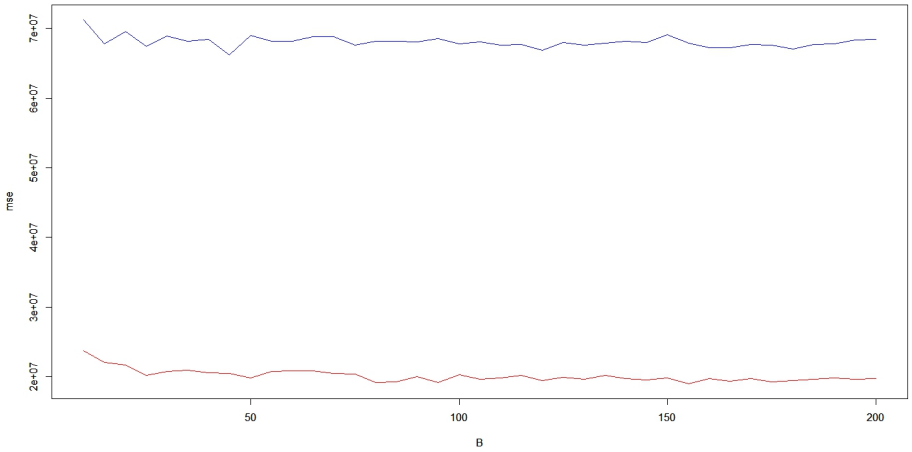

Figure 4: MSE vs. B. Blue is test set and red is training set.

c) In terms of testing error, the random forest with $B = 45$ performed the best for me. This yielded that the most undervalued player is Jack Cust with an actual salary of \$410,000, and a predicted salary of \$8,461,580. His salary through the years is shown in Table 2. As a general manager, as you can see that he only lasted 4 years after the 2008 season, the reason why he was undervalued was because he had an anomolous year.

Table 2: Salary of Jack Cust

Year	Salary
2008	\$410,000
2009	\$2,800,000
2010	\$2,650,000
2011	\$2,500,000
2012	\$148,000

Code Appendix

```
set.seed(525)

data <- read.csv("MLB2008.csv")
```

```
data.train <- data[1:154,]
data.test <- data[155:nrow(data),]

# Question 1a
library('rpart')
library('rpart.plot')

## Everything to the right of POS
x_names <- names(data)[6:134]
formula <- paste("SALARY ~ ", paste(x_names, collapse= "+"))

## No prune
data.rpart.model <- rpart(formula, data.train)
dev.new()
prp(data.rpart.model)

## MSE no prune
data.rpart.train.predict <- predict(data.rpart.model, data.train)
mse <- norm(data.rpart.train.predict - data.train$SALARY, type="2")
sprintf("MSE for trained data: %0.0f", mse)

data.rpart.test.predict <- predict(data.rpart.model, data.test)
mse <- norm(data.rpart.test.predict - data.test$SALARY, type="2")
sprintf("MSE for test data: %0.0f", mse)

## Prune parameter of 0.1
data.rpart.prune.model <- prune(data.rpart.model, cp=0.1)
dev.new()
prp(data.rpart.prune.model)

## MSE no prune
data.rpart.prune.train.predict <- predict(data.rpart.prune.model, data.
    train)
mse <- norm(data.rpart.prune.train.predict - data.train$SALARY, type="2"
    )
sprintf("MSE for trained data (pruned): %0.0f", mse)

data.rpart.prune.test.predict <- predict(data.rpart.prune.model, data.
    test)
mse <- norm(data.rpart.prune.test.predict - data.test$SALARY, type="2")
sprintf("MSE for test data (pruned): %0.0f", mse)

## Prune parameter of 0.3
data.rpart.prune.model <- prune(data.rpart.model, cp=0.3)
dev.new()
prp(data.rpart.prune.model)

## MSE no prune
data.rpart.prune.train.predict <- predict(data.rpart.prune.model, data.
    train)
mse <- norm(data.rpart.prune.train.predict - data.train$SALARY, type="2"
    )
sprintf("MSE for trained data (pruned): %0.0f", mse)

data.rpart.prune.test.predict <- predict(data.rpart.prune.model, data.
    test)
```

```
mse <- norm(data.rpart.prune.test.predict - data.test$SALARY, type="2")
sprintf("MSE for test data (pruned): %0.0f", mse)

# Question 1b
library('randomForest')

random_forest.mse.train <- data.frame(B=integer(), mse=numeric())
random_forest.mse.test <- data.frame(B=integer(), mse=numeric())

for(B in seq(10, 200, by=5)) {
    data.random_forest.model <- randomForest(as.formula(formula), data=
        data.train, ntree=B)
    df <- data.frame(B, mse=norm(predict(data.random_forest.model, data.
        train) - data.train$SALARY, type="2"))
    random_forest.mse.train <- rbind(random_forest.mse.train, df)
    df <- data.frame(B, mse=norm(predict(data.random_forest.model, data.
        test) - data.test$SALARY, type="2"))
    random_forest.mse.test <- rbind(random_forest.mse.test, df)
}

dev.new()
plot(random_forest.mse.train,type="l",col="red", ylim=range(c(random_
    forest.mse.train["mse"],random_forest.mse.test["mse"])))
par(new = TRUE)
plot(random_forest.mse.test,type="l",col="blue", ylim=range(c(random_
    forest.mse.train["mse"],random_forest.mse.test["mse"])))

# Question 1c

## Min MSE on test data is random forest with B=45 (predict on full data
     set)
data.random_forest.model <- randomForest(as.formula(formula), data=data.
    train, ntree=45)
data.random_forest.full <- predict(data.random_forest.model, data)

## Most undervalued
data.random_forest.undervalued <- (data.random_forest.full - data$SALARY
    )
undervalued <- which.max(data.random_forest.undervalued)
undervalued.player <- data[c("PLAYER","SALARY")][undervalued,]
sprintf("Most undervalued player: %s, Actual: $%0.0f, Predicted: $%0.0f"
    , undervalued.player$PLAYER, undervalued.player$SALARY, data.random_
    forest.undervalued[undervalued])
```

Exercise 2: In this problem, we will show an insane result in tree regression with categorical variables. We will prove that we can treat categorical variables as if they are ordered (imagine how counterintuitive it is for the zipcode data you used in the last homework).

Suppose we use the tree regression to fit the data $\{Y_i, X_i\}_{i=1}^n$. Here the predictor $\{X_i\}_{i=1}^n$ are categorical variables having M possible **unordered** values in $\{1, \ldots, M\}$. In order to minimize the square loss

$$\min_{f \in T_K} \sum_{i=1}^{n} (Y_i - f(X_i))^2$$

where $T_K = \{f : f(x) = \sum_{k=1}^{K} \alpha_k I(x \in L_k)\}$ is the tree function class with K leaves and L_k is a leaf. Since X_i's are categorical, the leaves L_k's are K-partition of M categories. Namely, L_k's are **nonempty** and **disjoint** sets such that $L_1 \cup \ldots \cup L_K = \{1, \ldots, M\}$.

Let the number of all possible partitions of $\{1, \ldots, M\}$ be $\mathcal{N}_1$. This is a huge number which makes the minimization above intractable. However, we will show in hte problem that the computation can be dramatically simplified by "pretending" X_i's to be ordered. Here is how we play the magic. For any $s \in \{1, \ldots, M\}$, without loss of generality, we assume the set $\{X_i = s\}$ is non-empty. Let

$$\bar{Y}_s = \frac{1}{|\{X_i = s\}|} \sum_{X_i = s} Y_i$$

Suppose $\bar{Y}_s$ are different for different s. Without loss of generality again, we assume that $\bar{Y}_1 < \bar{Y}_2 < \ldots < \bar{Y}_M$.

1) Prove that for $1 \leq u < v < w \leq M$, if $u, w \in L_k$ for some k, then $v \in L_k$ as well. (**Hint:** One possible way is to prove by contradiction. If $u, w \in L_k$ but $v \in L_{k'}$, show that at least one of the following lines is true:

$$|\bar{Y}_v - Avg(L_{k'})| \geq |\bar{Y}_v - Avg(L_k)| \text{ and } |\bar{Y}_v - Avg(L_{k'})| > 0$$
$$|\bar{Y}_u - Avg(L_k)| \geq |\bar{Y}_u - Avg(L_{k'})| \text{ and } |\bar{Y}_u - Avg(L_k)| > 0$$
$$|\bar{Y}_w - Avg(L_k)| \geq |\bar{Y}_w - Avg(L_{k'})| \text{ and } |\bar{Y}_w - Avg(L_k)| > 0$$

where $Avg(L_k)$ is the average of the set $\{Y_i : X_i \in L_k\}$. Then show that you can reorganize the partition such that the square loss can be smaller.)

2) Let $\mathcal{N}_2$ be the number of partitions we need to consider for the minimization above if we have proven Q2.1. Calculate the ratio $\mathcal{N}_2/\mathcal{N}_1$ for $K = 2$ and you can see how much we improve.

Answer:

1) Let us rewrite the minimization in terms of $Avg(L_k)$

$$\begin{aligned}
&\min_{f\in T_K} \sum_{i=1}^{n} (Y_i - f(X_i))^2 \\
&= \min_{f\in T_K} \sum_{i=1}^{n} Y_i^2 - 2Y_i f(X_i) + f(X_i)^2 \\
&= \min_{f\in T_K} \sum_{i=1}^{n} f(X_i)^2 - 2Y_i f(X_i) \\
&= \min_{\alpha_k} \sum_{k=1}^{K} \sum_{i=1}^{n} \left(\alpha_k^2 I(X_i \in L_k) - 2\alpha_k I(X_i \in L_k) Y_i\right) \\
&= \min_{\alpha_k} \sum_{k=1}^{K} \sum_{X_i \in L_k} \left(\alpha_k^2 - 2\alpha_k Y_i\right) \\
&= \min_{\alpha_k} \sum_{k=1}^{K} |\{L_k\}| \left(\alpha_k^2 - 2\alpha_k Avg(L_k)\right)
\end{aligned}$$

From here, it is evident that the minimizer is thus $\alpha_k = Avg(L_k)$. Thus, we have that the optimum is

$$\begin{aligned}
&\sum_{i=1}^{n} (Y_i - f(X_i))^2 \\
&= \sum_{k=1}^{K} \sum_{s\in L_k} \sum_{X_i = s} (Y_i - \alpha_k)^2 \\
&= \sum_{k=1}^{K} \sum_{s\in L_k} \sum_{X_i = s} \left((Y_i - \bar{Y}_s)^2 + 2(Y_i - \bar{Y}_s)(\bar{Y}_s - \alpha_k) + (\bar{Y}_s - \alpha_k)^2\right)
\end{aligned}$$

But we have that $\sum_{X_i=s} 2(Y_i - \bar{Y}_s)(\bar{Y}_s - \alpha_k) = 0$ so

$$
\begin{aligned}
&= \sum_{k=1}^{K} \sum_{s \in L_k} \sum_{X_i = s} \left((Y_i - \bar{Y}_s)^2 + (\bar{Y}_s - \alpha_k)^2 \right) \\
&= C(Y_i) + \sum_{k=1}^{K} \sum_{s \in L_k} |\{X_i = s\}| (\bar{Y}_s - \alpha_k)^2 \\
&= C(Y_i) + \sum_{k=1}^{K} \sum_{s \in L_k} |\{X_i = s\}| (\bar{Y}_s - Avg(L_k))^2
\end{aligned}
$$

Assume that $u, w \in L_k$ and $v \in L_{k'}$. We have that $Avg(L_k) \neq Avg(L_{k'})$ since this implies that $\alpha_k = \alpha_{k'}$ and so L_k and $L_{k'}$ are not disjoint. WLOG, let $Avg(L_k) < Avg(L_{k'})$. Now we treat three cases.

Case 1, $Avg(L_k) < Avg(L_{k'}) \leq \bar{Y}_v$. Since $\bar{Y}_w > \bar{Y}_v$, we have

$$
\begin{aligned}
0 < \bar{Y}_w - Avg(L_{k'}) &< \bar{Y}_w - Avg(L_k) \\
\implies |\bar{Y}_w - Avg(L_{k'})| &< |\bar{Y}_w - Avg(L_k)|
\end{aligned}
$$

Which contradicts the optimality of α_k.

Case 2, $Avg(L_k) \leq \bar{Y}_v < Avg(L_{k'})$. If $Avg(L_{k'}) < \bar{Y}_w$ then we get the same as above. Otherwise we have, $Avg(L_{k'}) \geq \bar{Y}_w$. By optimality of $Avg(L_{k'}$

$$
\begin{aligned}
\bar{Y}_v - Avg(L_k) &\geq Avg(L_{k'}) - \bar{Y}_v \\
2\bar{Y}_v &\geq Avg(L_{k'}) + Avg(L_k)
\end{aligned}
$$

But by assumption and optimality of $Avg(L_k)$

$$
2\bar{Y}_w \leq Avg(L_{k'}) + Avg(L_k)
$$

Which is a contradiction that $\bar{Y}_w > \bar{Y}_v$.

Case 3, $\bar{Y}_v < Avg(L_k) < Avg(L_{k'})$

$$
\begin{aligned}
0 < Avg(L_k) - \bar{Y}_v &< Avg(L_{k'}) - \bar{Y}_v \\
\implies |Avg(L_k) - \bar{Y}_v| &< |Avg(L_{k'}) - \bar{Y}_v|
\end{aligned}
$$

which contradicts the optimality of $Avg(L_{k'})$. This handles the three cases and so we conclude that $v \in L_k$ as well.

2) $\mathcal{N}_1 = 2^M - 1$ since that is how many non-empty partitions are possible. Since $K = 2$, we have that $\mathcal{N}_2 = M - 1$ since we only have to consider partitions that have the property $1 \leq u < v < w \leq M$, if $u, w \in L_k$ for some k, then $v \in L_k$ as well. This can be seen by the fact that when $K = 2$, we must divided $\{1, \ldots, M\}$ into two sets with the largest element of the first set smaller than the least element of the second set. That is, they are of the form $\{1, \ldots, i\}$ and $\{i + 1, \ldots, M\}$. There are only $M - 1$ choices for i. Thus, the ratio is $\mathcal{N}_2/\mathcal{N}_1 = \frac{M-1}{2^M-1}$.

Exercise 3: In the problem, we will provide a theoretical justification of the dropout trick we learned in depp learning lectures. We will show that dropout is essentially a regularization. This problem also gives us a new statistical method to implement regularization.

Consider the generalized linear models (GLMs), which defines a conditional distribution over a response $y \in \mathcal{Y}$ given an input feature vecture $x \in \mathbb{R}$.

$$p_\beta(y|x) := h(y)e^{y \cdot x^T\beta - A(x^T\beta)}$$

Here $h(y)$ is a quantity independent of x and β, $A(\cdot)$ is the log-partition function which makes $p_\beta(y|x)$ a valid conditional distribution. Define the negative log-likelihood to be $\ell_{x,y}(\beta) := -\log p_\beta(y|x)$. Instead of x_i, we consider the noisy version $\tilde{x}_i = \nu(x_i, \xi_i)$, where two types of noise are considered:

- **Additive Gaussian noise:** $\nu(x_i, \xi_i) = x_i = \xi_i$, where $\xi_i \sim \mathcal{N}(0, \sigma^2 I_{d\times d})$
- **Dropout noise:** $\nu(x_i, \xi_i) = x_i \odot \xi_i$, where $\odot$ is the element-wise product of two vectors and each component of ξ_i is an independent Bernoulli draw, i.e. $\xi_{ij} = 1/(1-\delta)$ with probability $1-\delta$ and 0 otherwise for $1 \le j \le d$.

Integrating over the feature noise gives us a noised maximum likelihood parameter estimate:

$$\hat{\beta} := \arg\min_{\beta\in} \sum_{i=1}^{n} \mathbb{E}_\xi[\ell_{\tilde{x}_i, y_i}(\beta)]$$

where the expectation $\mathbb{E}_\xi[\cdot]$ is taken with respect to $\xi = (\xi_1, \ldots, \xi_n)$ only.

1) Prove that as long as $\mathbb{E}[\tilde{x}] = x$, we have $\sum_{i=1}^n \mathbb{E}_\xi[\ell_{\tilde{x}_i,y_i}(\beta)]$ $= \sum_{i=1}^n \ell_{x_i,y_i}(\beta) + R(\beta)$, where

$$R(\beta) = \sum_{i=1}^{n} \big(\mathbb{E}[A(\tilde{x}_i^T\beta)] - A(x_i^T\beta)\big)$$

2) If $\mathbb{E}_\xi[\tilde{x}] = x$ and A''' is bounded, show that

$$\mathbb{E}_\xi[A(\tilde{x}^T\beta)] - A(x^T\beta) = \frac{1}{2}A''(x^T\beta)Var_\xi(\tilde{x}^T\beta) + O(\mathbb{E}_\xi[|(\tilde{x}-x)^T\beta|^3])$$

3) Define $R^q(\beta) := \frac{1}{2}\sum_{i=1}^n A''(x_i^T\beta)Var_\xi(\tilde{x}_i^T\beta)$, show that in the linear regression setting, we have

- $R^q(\beta) = \frac{1}{2}\sigma^2 n\|\beta\|_2^2$ when using the additive Gaussian noise,
- $R^q(\beta) = \frac{1}{2}\frac{\delta}{1-\delta}\beta^T\text{diag}(X^TX)\beta$ using dropout noise, where $X \in \mathbb{R}^{n\times d}$ is the design matrix.

Answer:

1) We have that

$$
\begin{aligned}
&\sum_{i=1}^n \mathbb{E}_\xi[\ell_{\tilde{x}_i,y_i}(\beta)] \\
&= \sum_{i=1}^n \mathbb{E}_\xi[-\log p_\beta(y_i|\tilde{x}_i)] \\
&= -\sum_{i=1}^n \mathbb{E}_\xi[\log\left(h(y_i)e^{y_i\cdot\tilde{x}_i^T\beta - A(\tilde{x}_i^T\beta)}\right)] \\
&= -\sum_{i=1}^n \left(\log h(y_i) + \mathbb{E}_\xi[y_i\cdot\tilde{x}_i^T\beta] - \mathbb{E}_\xi[A(\tilde{x}_i^T\beta)\right)] \\
&= -\sum_{i=1}^n \left(\log h(y_i) + y_i\cdot\mathbb{E}_\xi[\tilde{x}_i^T]\beta - A(x_i^T\beta)\right) + A(x_i^T\beta) - \mathbb{E}_\xi[A(\tilde{x}_i^T\beta)] \\
&= -\sum_{i=1}^n \left(\log h(y_i) + y_i\cdot x_i^T\beta - A(x_i^T\beta)\right) + \sum_{i=1}^n \left(\mathbb{E}_\xi[A(\tilde{x}_i^T\beta)] - A(x_i^T\beta)\right) \\
&= \sum_{i=1}^n \ell_{x_i,y_i}(\beta) + R(\beta)
\end{aligned}
$$

2) We do a Taylor expansion of A at $x\beta$

$$
\begin{aligned}
&A(\tilde{x}^T\beta) \\
&= A(x^T\beta) + A'(x^T\beta)(\tilde{x}-x)^T\beta + \frac{1}{2}A''(x^T\beta)\beta^T(\tilde{x}-x)^2\beta + O(|(\tilde{x}-x)^T\beta|^3)
\end{aligned}
$$

Taking expectations and recalling that x is not random

$$\begin{aligned}
\mathbb{E}_\xi[A(\tilde{x}^T\beta)] &= \mathbb{E}_\xi[A(x^T\beta)] + \mathbb{E}_\xi[A'(x^T\beta)(\tilde{x}-x)^T\beta] \\
&\quad + \frac{1}{2}\mathbb{E}_\xi[A''(x^T\beta)\beta^T(\tilde{x}-x)^2\beta] \\
&\quad + O(\mathbb{E}_\xi[|(\tilde{x}-x)^T\beta|^3]) \\
\mathbb{E}_\xi[A(\tilde{x}^T\beta)] &= A(x^T\beta) + A'(x^T\beta)\mathbb{E}_\xi[(\tilde{x}-x)^T]\beta \\
&\quad + \frac{1}{2}A''(x^T\beta)\mathbb{E}_\xi[\beta^T(\tilde{x}-\mathbb{E}_\xi[x])^2\beta] \\
&\quad + O(\mathbb{E}_\xi[|(\tilde{x}-x)^T\beta|^3]) \\
\mathbb{E}_\xi[A(\tilde{x}^T\beta)] - A(x^T\beta) &= \frac{1}{2}A''(x^T\beta)Var(\tilde{x}^T\beta) + O(\mathbb{E}_\xi[|(\tilde{x}-x)^T\beta|^3])
\end{aligned}$$

Where the first derivative term disappears because $\mathbb{E}_\xi[\tilde{x}] = x$.

3) Since we are in the linear regression setting, we have that $y_i \sim \mathcal{N}(0,1)$. So $A(x_i^T\beta) = (x_i^T\beta)^2/2$. So $A''(x^T\beta) = 1$. Now, let us compute the variance for the two settings. For additive Gaussian noise:

$$\begin{aligned}
Var_\xi(\tilde{x}_i^T\beta) &= \beta^T Var_\xi(x_i^T + \xi_i)\beta \\
&= \beta^T Var_\xi(\xi_i)\beta \\
&= \beta\sigma^2 I_{d\times d}\beta \\
&= \sigma^2\|\beta\|_2^2
\end{aligned}$$

Where the second line is from the fact that x_i is fixed. For dropout:

$$\begin{aligned}
\sum_{i=1}^{d} Var_\xi(\tilde{x}_i^T\beta) &= \sum_{i=1}^{d} \beta^T Var_\xi(x_i^T \odot \xi_i)\beta \\
&= \sum_{i,j=1}^{d} \beta_j Var_\xi(x_{i_j} \cdot \xi_{i_j})\beta_j = \sum_{i,j=1}^{d} \beta_j x_{i_j}^2 Var_\xi(\xi_{i_j})\beta_j \\
&= \sum_{i,j=1}^{d} \beta_j x_{i_j}^2 \left(\frac{1}{1-\delta} - 1\right)\beta_j = \frac{\delta}{1-\delta}\sum_{i,j=1}^{d} \beta_j x_{i_j}^2 \beta_j \\
&= \frac{\delta}{1-\delta}\beta^T \text{diag}(X^T X)\beta
\end{aligned}$$

Where we use the fact that the Bernoulli's are independent to ignore the cross variance terms on the second line. We conclude that

$$R^q(\beta) = \frac{1}{2}\sigma^2 n\|\beta\|_2^2$$

for additive Gaussian noise and

$$R^q(\beta) = \frac{1}{2}\frac{\delta}{1-\delta}\beta^T \text{diag}(X^T X)\beta$$

for dropout.

Exercise 4: In this problem, we will explore the properties of RIP. We say a matrix A is restricted isometry for s, if there exists $\delta \in [0, 1)$, such that

$$(1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2$$

for any $\|x\|_0 \leq s$, where $\|x\|_0 = |\text{supp}(x)|$.

1) For any $1 \leq s \leq d$, the matrix $A \in \mathbb{R}^{m\times d}$ satisfies the RIP with parameter δ. Prove that

$$|\langle Ax, Ay\rangle| \leq \delta_{s+t}\|x\|_2\|y\|_2$$

for any $\|x\|_0 \leq s$, $\|y\|_0 \leq t$ and x, y have disjoint supports (**Hint:** Use the polarization identity).

2) In the class, we show the perfect recovery of compressed sensing under $3s$-RIP condition. In this problem, we will improve the result. We will show that $2s$-RIP is actually enough for perfect recovery. In specific, prove that if X satisfies $2s$-RIP with $\delta_{2s} \leq 1/(1 + \sqrt{2})$ and $\|\beta^*\|_0 \leq s$, then $\hat{\beta} = \beta^*$ where

$$\hat{\beta} = \arg\min_{\beta} \|\beta\|_1 \text{ s.t. } X\beta^* = X\beta$$

Think about why we did not get a sharp result in the class?

(**Hint:** Instead of dividing the $h = \hat{\beta} - \beta^*$ into parts $S^*, S_1, S_2, \ldots$ of sizes $2s$ in the class, try s. Then show $\|h_{S^*}\|_1 \leq \rho\|h_{S^{*c}}\|_1$ for some $\rho < 1$ by starting with

$$\|Xh_{S^*\cup S_1}\|_2^2 = -\sum_{j\geq 2}\langle Xh_{S^*\cup S_1}, Xh_{S_j}\rangle$$

and apply Q4.1).

3) In the class, we show a sufficient condition for the RIP condition. This problem will give a lower bound of samples needed for the RIP condition. Suppose $A \in \mathbb{R}^{n\times d}$ satisfy RIP with $\delta_{2s} \in (0, 1/2]$. Prove that

$$n \geq Cs \log\left(\frac{d}{s}\right)$$

for some constant C independent of d, n, and s. You can prove this problem following the steps below.

Step 1: Let s and d satisfying $s < d/2$ be given. Prove that there exists a set

$$\mathcal{X} \subseteq U = \{x \in \{0, +1, -1\}^d : \|x\|_0 = s\}$$

such that for any $x, z \in \mathcal{X}$ with $x \neq z$ we have

$$\|x - z\|_2 \geq \sqrt{s/2} \text{ and } \log|\mathcal{X}| \geq \frac{s}{2} \log\left(\frac{d}{s}\right)$$

You can construct your set by adding points in U one by one and make sure there are points z's that have not been added so far satisfying $\|x - z\|_2 \geq \sqrt{s/2}$ for x's that have already been added. You need to bound the number of these possible z's if m of x's have been added until you find that it is impossible to find another such z. Then you can find the lower bound of $|\mathcal{X}|$.

Step 2: For any pair of $x, z \in \mathcal{X}$, the balls centered at Ax, Az with radius $\sqrt{s/16}$ are disjoint.

Step 3: Prove the problem using Step 1 and Step 2. (**Hint:** Find a larger ball containing all balls in Step 2. Use the fact that the volume of this larger ball is bigger than the sum of the smaller balls in Step 2.)

4) Let $X \in \mathbb{R}^{n \times d}$. Here we consider three different methods for generating X: i.i.d. Gaussian (via `Gaussian_Phi.m`), random rows of a discrete cosine transform (via `SubDCT_Phi.m`), and consecutive rows of a random Toeplitz matrix (via `SubToep_Phi.m`). Let β be a vector s.t. $\|\beta\|_0 = s$ and $Y = X\beta$. Define the compressed sensing estimator

$$\hat{\beta} = \arg\min_{\beta} \|\beta\|_1, \text{ s.t. } Y = X\beta$$

You can solve the prboblem via `l1eq_pd.m`. In each of these three cases, set $d = 1024$ and use simulation to determine values of n and s such that we can recover the true β. (You can create sparse vectors β at random for the simulation using any distribution you find reasonable). In specific, for each pair of (n, s), use Monte Carlo methods to calculate the probability that the true β is recoverd.

You can visualize your results using a plot where the x-axis is s/d and y-axis is n/d and the rainbow color represents the probability for exact recovery. Do your plots imply some connection to Q4.3 and random matrix theory for RIP? (This is an open question and you can explore as much as possible).

Answer:

1) Since x and y have disjoint supports we have that

$$\|x \pm y\|_2^2 = \|x\|_2^2 + \|y\|_2^2$$
$$\|x \pm y\|_0 = \|x\|_0 + \|y\|_0 \leq s + t$$

This gives us the restricted isometry inequalties

$$(1 - \delta_{s+t})\|x \pm y\|_2^2 \leq \|Ax \pm Ay\|_2^2 \leq (1 + \delta_{s+t})\|x \pm y\|_2^2$$

Now we use the polarization identity coupled with the above to get

$$\langle Ax, Ay\rangle = \frac{1}{4}(\|Ax + Ay\|_2^2 - \|Ax - Ay\|_2^2)$$
$$\langle Ax, Ay\rangle \leq \frac{1}{4}((1 + \delta_{s+t})\|x + y\|_2^2 - (1 - \delta_{s+t})\|x - y\|_2^2)$$
$$|\langle Ax, Ay\rangle| \leq \frac{1}{2}\delta_{s+t}(\|x\|_2^2 + \|y\|_2^2)$$

Now we use the fact that we can assume that x and y have the same 2 norm without loss of generality (since we can just divide the above inequality by $\|x\|_2\|y\|_2$) to get the result $|\langle Ax, Ay\rangle| \leq \delta_{s+t}\|x\|_2\|y\|_2$.

2) We follow the hint and the class proof for $3s$-RIP. From class, the proof that $\|h_{S^{*c}}\|_1 \leq \|h_{S^*}\|_1$ is unchanged. Now, we divide $h = \hat{\beta} - \beta$ into parts $S_0 = S^*$, $S_1 =$ locations of s-largest entries in $h_{S^{*c}}$, $S_2 =$ next s largest locations, etc. Then we have

$$\|Xh_{S^* \cup S_1}\|_2^2 = -\sum_{j \geq 2} \langle Xh_{S^* \cup S_1}, XH_{S_j}\rangle$$
$$\iff \|Xh_{S^* \cup S_1}\|_2^2 \leq \sum_{j \geq 2} |\langle Xh_{S^* \cup S_1}, XH_{S_j}\rangle|$$

We now upper bound the right hand side using Q4.1 and lower bound the LHS using $2s$-RIP

$$(1-\delta_{2s})\|h_{S^*\cup S_1}\|_2^2 \leq \sum_{j\geq 2}\delta_{s+s}\|h_{S^*\cup S_1}\|_2\|h_{S_j}\|_2$$
$$\iff \|h_{S^*\cup S_1}\|_2 \leq \frac{\delta_{2s}}{1-\delta_{2s}}\sum_{j\geq 2}\|h_{S_j}\|_2$$

Now, we upper and lower bound the 2 norms. Lower bound first

$$\|h_{S^*\cup S_1}\|_2 \geq \|h_{S^*}\|_2 \geq \frac{\|h_{S^*}\|_1}{\sqrt{s}}$$

Now upper bound

$$\frac{\delta_{2s}}{1-\delta_{2s}}\sum_{j\geq 2}\|h_{S_j}\|_2 \leq \frac{\delta_{2s}}{1-\delta_{2s}}\sum_{j\geq 2}\sqrt{s}\|h_{S_j}\|_\infty \leq \frac{\delta_{2s}}{1-\delta_{2s}}\sum_{j\geq 2}\frac{\|h_{S_j}\|_1}{\sqrt{s}}$$

So we conclude

$$\frac{\|h_{S^*}\|_1}{\sqrt{s}} \leq \frac{\delta_{2s}}{1-\delta_{2s}}\sum_{j\geq 2}\frac{\|h_{S_j}\|_1}{\sqrt{s}}$$
$$\implies \|h_{S^*}\|_1 \leq \frac{\delta_{2s}}{1-\delta_{2s}}\|h_{S^{*c}}\|_1$$

Where $\rho = \frac{\delta_{2s}}{1-\delta_{2s}} < 1$ by the conditions on δ_{2s}.

3) Let us show that one needs $n \geq Cs\log\left(\frac{d}{s}\right)$ samples for the RIP condition.

Step 1: First, since $\mathcal{X} \subseteq U$, we have that $|\mathcal{X}| \leq |U|$. So, we must find a way to lower ound $|\mathcal{X}|$. Let $\mathcal{X}$ be a subset of U with $\|x-z\|_2 \geq \sqrt{s/2}$ for $x, z \in \mathcal{X}$. Such a set exists because one can take a single point. Take $\mathcal{X}$ to be the largest possible set. Thus, we have $U = \cup_{x\in\mathcal{X}} B_x(\sqrt{s/2})$. Where $B_x(\sqrt{s/2})$ is the ball around x of radius $\sqrt{s/2}$. Thus,

$$|U| \leq \sum_{x\in\mathcal{X}}|B(x,\sqrt{s/2})| \leq |\mathcal{X}||B_0(\sqrt{s/2})|$$

Where the above can be visualized that we have points in $\mathcal{X}$ and that all the balls of radius $\sqrt{s/2}$ around these points contain U by definition.

Furthermore, the ball centered at 0 contains the most points in U and so we can upper bound any ball with this one. Now, we have

$$\begin{aligned}|B_0(\sqrt{s/2})| &= |\{y \in U : \|0-y\|_2 \leq \sqrt{s/2}\}| \\ &\leq |\{y \in U : \|y\|_0 \leq \sqrt{s/2}\}| = \binom{d}{\frac{s}{2}} 3^{\frac{s}{2}}\end{aligned}$$

But we also have that the total points in U is $\binom{d}{s}2^s$. This is due to the fact that there must be s non zero entries and you have a choice of ± 1 for these points. Thus, putting these together,

$$\begin{aligned}|\mathcal{X}| &\geq \left(\frac{4}{3}\right)^{\frac{s}{2}} \frac{\binom{d}{s}}{\binom{d}{\frac{s}{2}}} \\ \implies |\mathcal{X}| &\geq \left(\frac{4}{3}\right)^{\frac{s}{2}} \frac{(d/s)^s}{(2d/s)^{s/2}} \\ \implies |\mathcal{X}| &\geq \left(\frac{2}{3}\right)^{\frac{s}{2}} (d/s)^{s/2} \\ \implies \log|\mathcal{X}| &\geq \frac{s}{2}\log(2/3) + \frac{s}{2}\log\left(\frac{d}{s}\right) \\ \implies \log|\mathcal{X}| &\geq \frac{s}{2}\log\left(\frac{d}{s}\right)\end{aligned}$$

Step 2: We have that $x, z \in \mathcal{X}$ then $\|x-z\|_2 \geq \sqrt{s/2}$ and $\|x-z\|_0 \leq 2s$. That is, we can use the results from Q4.1 to get

$$\|Ax - Az\|_2^2 \geq (1-\delta_{2s})\|x-z\|_2^2 \geq (1-\delta_{2s})\frac{s}{2} \geq \frac{s}{4}$$

Where we $\delta_{2s} \in [0, 1/2]$ by assumption. Thus, we have that Ax and Az are separated by more than $\sqrt{s/4}$ and we conclude that this is twice the radii.

Step 3: We wish to contain all the disjoint balls in Step 2. Thus, we use that for $x \in U$ thath $\|Ax\|_2^2 \leq (1+\delta_s)\|x\|_2^2 \leq \frac{3}{2}\|x\|_0 = \frac{3}{2}s$. Thus, to contain the balls in Step 2, we must add $1/4\sqrt{s}$ to the radius of the ball centered at the origin with radius $\sqrt{3/2}\sqrt{s}$. Thus, the larger ball is centered at the origin with radius ($\sqrt{25s/16}$. Now we have that the

sum of the small balls in Step 2 is less than the volume of this larger ball

$$\sum_{x \in \mathcal{X}} Vol(\sqrt{s/16}) \leq Vol(\sqrt{25s/16})$$
$$\implies |\mathcal{X}| \sqrt{s/16}^n Vol(1) \leq \sqrt{25s/16}^n Vol(1)$$
$$\implies \frac{s}{2} \log\left(\frac{d}{s}\right) \leq n \log(5)$$
$$\implies s \log\left(\frac{d}{s}\right) \leq n \log(25)$$
$$\implies Cs \log\left(\frac{d}{s}\right) \leq n$$

4) For computation reasons, I have restricted d to 512 and used a step size of 50 for n and s. Also, 10 iterations of Monte Carlo was done. The figures below show the probabilities for the different random matrices. The x-axis is s/d, y-axis is n/d, and red is probability 1, blue is probability 0. Note that the y-axis is inverted. You can see the relation to question Q4.3 since $\frac{n}{d} \geq C\frac{s}{d} \log\left(\frac{s}{d}\right)$ for the RIP condition and compare that to the frontier where the probabilities transition from 0 to 1. The code is appended below.

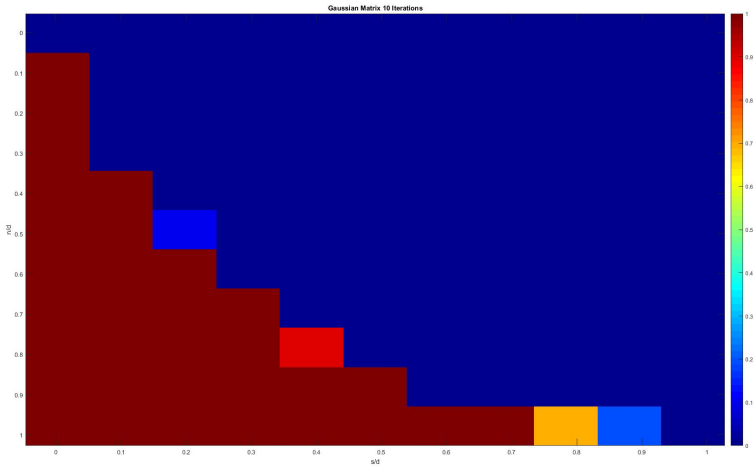

Figure 5: Gaussian Matrix Probabilities

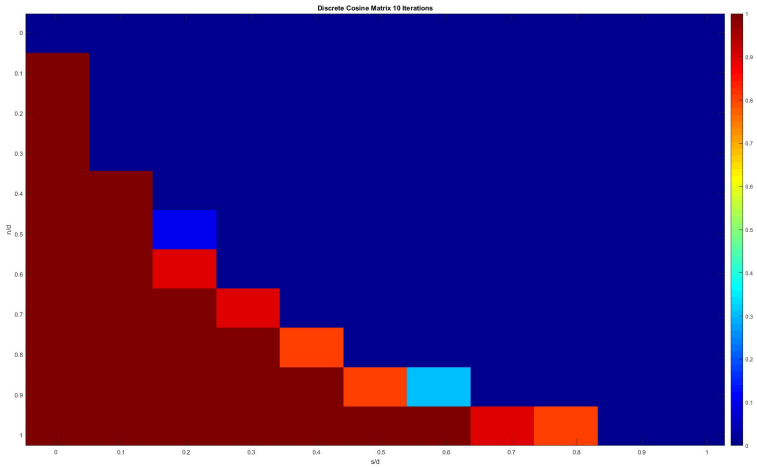

Figure 6: Discrete Cosine Transform Matrix Probabilities

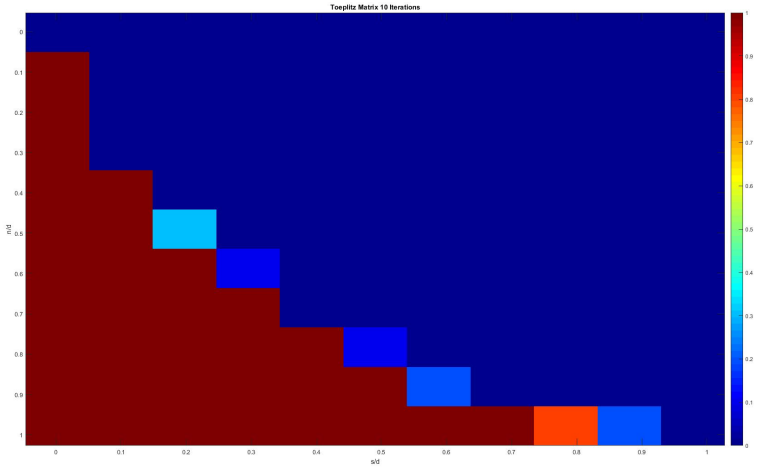

Figure 7: Toeplitz Matrix Probabilities

Code Appendix

```
clear;
clc;

d = 512;
```

```
iterations = 10;
% Keep track of probability of success for pair (n,s)
success = zeros(floor(d/50)+1, floor(d/50)+1);

for i=1:iterations
    for s=1:50:d
        for n=1:50:d
            % Change the random method
            X = SubDCT_Phi(n,d);

            % Generate random beta and zero the first d-s
                entries
            beta = randn(d, 1);
            beta(1:(d-s),1)=0;

            Y = X*beta;

            sol = l1eq_pd(0*beta, X, 0*X, Y);

            % Check if solution is close to original beta
            if(norm(l1eq_pd(0*beta, X, 0*X, Y) - beta) < 0.01)
                success(floor(n/50) + 1, floor(s/50) + 1) =
                    success(floor(n/50) + 1, floor(s/50) + 1)
                    + 1;
            end
        end
    end
end

dlmwrite('dct.txt', success/iterations)
colormap('jet')
imagesc((1:50:d)/d,(1:50:d)/d, success/iterations)
colorbar
xlabel('s/d')
ylabel('n/d')
title('Discrete Cosine Matrix 10 Iterations')
```