MO444/MC886 - Aprendizado de Máquina e Reconhecimento de Padrões

Raí Caetano de Jesus ———— RA 188971

Exercício 1

1

Faça o PCA dos dados (sem a última coluna).

```
#lê o dataset
data = data.matrix(read.csv("data1.csv", header = TRUE));

#armazena a ultima coluna do dataset
classe = data[,ncol(data)];

#remove os cabeçalhos do dataset
dimnames(data) = NULL;

#dados originais sem a última coluna
dados = data[,1:ncol(data)-1];

#faz o PCA
pca = prcomp(dados, scale. = T);
```

Se voce quiser que os dados transformados tenham 80% da variância original, quantas dimensões do PCA você precisa manter?

```
#faz a soma acumulativa para descobrir quantas colunas do PCA manter
soma_acum = cumsum(pca$sdev^2)/sum(pca$sdev^2);

#pega a quantidade de colunas necessárias
num_cols <- which(soma_acum >= 0.8)[1];
cat('Quantidade de dimensões necessárias para manter uma variância de 80%: ', num_cols);
```

```
## Quantidade de dimensões necessárias para manter uma variância de 80%:  13
```

Gere os dados transformados mantendo 80% da variância. (Atenção, este passo não é 100% correto do ponto de vista de aprendizado de maquina. Não repita este passo em outras atividades).

```
#gera os dados transformados mantendo 80% da variância
dados_transformados = pca$x[,1:num_cols];
```

Considere as primeiras 200 linhas dos dados como o conjunto de treino, e as 276 ultimas como o conjunto de dados

```
#conjunto de treino com PCA
t_set_PCA = data.frame(dados_transformados[1:200,]);
#adiciona a coluna com a classe
t_set_PCA = as.data.frame(cbind(t_set_PCA, classe[1:200]));
#nomeia a coluna de "classe"
colnames(t_set_PCA)[ncol(t_set_PCA)] = "classe";

#conjunto de teste com PCA
d_set_PCA = data.frame(dados_transformados[201:nrow(dados_transformados),]);
d_set_PCA = as.data.frame(cbind(d_set_PCA, classe[201:length(classe)]));
colnames(d_set_PCA)[ncol(d_set_PCA)] = "classe";

#conjuntos de treino sem PCA (dados originais)
t_set = data.frame(dados[1:200,1:num_cols]);
t_set = as.data.frame(cbind(t_set, classe[1:200]));
colnames(t_set)[ncol(t_set)] = "classe";

#conjuntos de treino sem PCA (dados originais)
d_set = data.frame(dados[201:nrow(dados),1:num_cols]);
d_set = as.data.frame(cbind(d_set, classe[201:length(classe)]));
colnames(d_set)[ncol(d_set)] = "classe";
```

2

Treine uma regressão logística no conjunto de treino dos dados originais e nos dados transformados.

```
#realiza a regressão logística usando o conjunto de treino com PCA
result_lr_PCA = glm(classe~., family=binomial(link="logit"), data = t_set_PCA);

#prevê um novo modelo de dados utilizando o conjunto de teste com PCA
new_model_PCA = predict.glm(result_lr_PCA, d_set_PCA, type = "response");

#realiza a regressao logista usando o conjunto de treino sem PCA
result_lr = glm(classe~., family=binomial(link="logit"), data = t_set);

#prevê um novo modelo de dados utilizando o conjunto de teste sem PCA
new_model = predict.glm(result_lr, d_set, type = "response");

#classifica o modelo com PCA previsto após a regressão logística
y_PCA = ifelse(new_model_PCA >= 0.5,1,0);

#classifica o modelo sem PCA previsto após a regressão logística
y = ifelse(new_model >= 0.5,1,0);
```

Qual a taxa de acerto no conjunto de teste nas 2 condições (sem e com PCA)?

```
#calcula accuracy com PCA
accuracy_lr_PCA = (sum(y_PCA == d_set_PCA$classe) / length(y_PCA)) * 100;

#calcula accuracy sem PCA
accuracy_lr = (sum(y == d_set$classe) / length(y)) * 100;
```

```
#imprimi acurácias
cat('Acurácia com PCA: ', accuracy_lr_PCA, '%\nAcurácia sem PCA: ', accuracy_lr, '%');
```

```
## Acurácia com PCA:  79.34783 %
## Acurácia sem PCA:  63.76812 %
```

3

Treine o LDA nos conjuntos de treino com e sem PCA e teste nos respectivos conjuntos de testes.

```
#realiza a LDA usando o conjunto de treino com PCA
result_lda_PCA = lda(classe~., data = t_set_PCA);

#realiza a LDA usando o conjunto de treino sem PCA
result_lda = lda(classe~., data = t_set);

#prevê um novo modelo de dados utilizando o conjunto de teste com PCA
new_model_PCA = predict(result_lda_PCA, d_set_PCA);

#prevê um novo modelo de dados utilizando o conjunto de teste sem PCA
new_model = predict(result_lda, d_set);

#classifica o modelo com PCA previsto após a LDA
y_PCA = ifelse(new_model_PCA$x >= 0.0,1,0);

#classifica o modelo sem PCA previsto após a LDA
y = ifelse(new_model$x >= 0.0,1,0);
```

Qual a acurácia nas 2 condições?

```
#calcula accuracy com PCA
accuracy_lda_PCA = (sum(y_PCA == d_set_PCA$classe) / length(y_PCA)) * 100;

#calcula accuracy sem PCA
accuracy_lda = (sum(y == d_set$classe) / length(y)) * 100;

#imprimi acurácias
cat('Acurácia com PCA: ', accuracy_lda_PCA, '%\nAcurácia sem PCA: ', accuracy_lda, '%');
```

```
## Acurácia com PCA:  77.53623 %
## Acurácia sem PCA:  61.95652 %
```

OBS: No exemplo de LDA, o objeto retornado pelo método predict() contém o atributo “class”, o qual é a classificação calculada pelo método. Ao calcular a acurácia comparando-se os dados do treino com este atributo, o resultado difere do calculado a mão, provavelmente devido a comparação utilizada no if (se >= 0 recebe 1, do contrário recebe 0). O valor utilizado para comparação provavelmente deve ser outro para que os resultados batam.

4

Qual a melhor combinação de classificador e PCA ou não?

Acurácia

```
cat('Acurácia:\n\n',
    '\t\t\t\t', 'LR', '\t\t', 'LDA', '\n',
    'com PCA', '\t', accuracy_lr_PCA, '%\t ', accuracy_lda_PCA, '%\n',
    'sem PCA', '\t', accuracy_lr, '%\t ', accuracy_lda, '%\n');
```

```
## Acurácia:
##
##                    LR          LDA
##  com PCA      79.34783 %   77.53623 %
##  sem PCA      63.76812 %   61.95652 %
```

Com base na Acurácia, o melhor classificador foi:

```
cat('Com PCA: ', ifelse(accuracy_lr_PCA > accuracy_lda_PCA, 'LR',
           ifelse(accuracy_lr_PCA < accuracy_lda_PCA, 'LDA', 'Iguais')),
    '\nSem PCA: ', ifelse(accuracy_lr > accuracy_lda, 'LR',
           ifelse(accuracy_lr < accuracy_lda, 'LDA', 'Iguais')));
```

```
## Com PCA:  LR
## Sem PCA:  LR
```