Тема 3. Методы машинного обучения. Методы отбора признаков по типу «обертки». Метод k ближайших соседей

Метод K ближайших соседей

В случае использования метода K ближайших соседей объекту присваивается класс, к которому относится большинство из K объектов обучающей выборки, наиболее похожих на него по признакам. Для оценки «сходства» или «несходства» объектов в пространстве отобранных признаков используют различные метрики, наиболее часто – евклидово расстояние (расстояние между объектами по прямой). Целью обучение является подбор оптимального значения K. Вначале построим соответствующую модель, используя методы фильтрации признаков (см. тему 2).

Импортируем таблицу, содержащую данные по 99 образцам рака молочной железы, экспрессирующим и неэкспрессирующим эстрогеновые рецепторы, из файла «Breast cancer 2.txt». Таблица содержит информацию по уровню транскрипции 10273 генов человека, которые будут использоваться в качестве признаков для классификации образцов. Разделим данные на обучающую и тестовую выборки:

```
cancer<-read.delim("Breast cancer 2.txt")
inTr<-createDataPartition(cancer$Class,p=0.7,list=FALSE)
tr<-cancer[inTr,]
test<-cancer[-inTr,]
```

Для удобства сохраним в отдельные переменные столбец с меткой класса и признаки:

```
tr.x<-tr[,-1]
tr.y<-tr[,1]
test.x<-test[,-1]
test.y<-test[,1]
```

Удалим «лишние» признаки, которые сильно коррелируют друг с другом:

```
cr<-cor(tr.x, method="spearman")
i<-findCorrelation(cr, cutoff=0.9)
tr.x<-tr.x[, -i]
test.x<-test.x[, -i]
```

После этого выполним тест Стьюдента для каждого столбца таблицы с обучающей выборкой, предварительно конвертировав таблицу данных в матрицу, потом отсортируем таблицу с результатами по возрастанию величины p:

```
tt<-colttests(as.matrix(tr.x), tr.y)
tt<-tt[order(tt$p.value),]
```

Функция **order** возвращает исходные индексы строк в новом порядке, заданном сортировкой.

Вначале отберем признаки, значения которых статистически значимо отличаются между классами (значение $p < 0.05$):

```
sign.gene<-rownames(tt)[tt$p.value<0.05]
```

Однако количество отобранных признаков слишком велико (2285). Отберем первые 500 признаков для построения модели:

```
sign.gene<-rownames(tt)[1:500]
```

Оставим в обучающей и тестовой выборках только отобранные признаки:

```
tr.x1<-tr.x[ , sign.gene]
test.x1<-test.x[ , sign.gene]
```

Важно отметить, что отбор признаков при помощи теста Стьюдента был выполнен именно для обучающей выборки, поскольку тестовая выборка должна оставаться полностью независимой от всех вычислений, которые производятся при построении классификационной модели, включая и отбор признаков.

Построим соответствующую модель при помощи функции **train**, которая подберет оптимальное число соседей:

```
fit<-train(x=tr.x1, y=tr.y, method="knn", trControl=trainControl(method="cv", number=10))
```

```
> fit

k-Nearest Neighbors

 70 samples
500 predictors
  2 classes: 'ER(-)', 'ER(+)'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 5 times)
Summary of sample sizes: 55, 57, 56, 56, 56, 55, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.8857582  0.7341993
  7  0.9058168  0.7831759
  9  0.9113407  0.7953772

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```

Точность прогноза, вычисленная в ходе пятикратного скользящего контроля, при значении K = 9 составляет 0.911. Выполним прогноз для тестовой выборки:

```
> pred<-predict(fit,newdata=test.x1)
> confusionMatrix(data=pred,reference=test.y,positive="ER(+)")
```

```
Confusion Matrix and Statistics

          Reference
Prediction ER(-) ER(+)
     ER(-)     5     2
     ER(+)     4    18

               Accuracy : 0.7931
                 95% CI : (0.6028, 0.9201)
    No Information Rate : 0.6897
    P-Value [Acc > NIR] : 0.1576

                  Kappa : 0.4852

 Mcnemar's Test P-Value : 0.6831

            Sensitivity : 0.9000
            Specificity : 0.5556
         Pos Pred Value : 0.8182
         Neg Pred Value : 0.7143
             Prevalence : 0.6897
         Detection Rate : 0.6207
   Detection Prevalence : 0.7586
      Balanced Accuracy : 0.7278

       'Positive' Class : ER(+)
```

Методы отбора признаков, основанные на «обертке»

В качестве примера рассмотрим два метода отбора признаков по типу «обертки»: метод рекурсивного исключения признаков и метод имитации отжига. Оба метода реализованы в функциях **rfe** и **safs** из пакета **caret**, которые позволяют строить модели одновременно с отбором признаков.

Отберем признаки с помощью метода рекурсивного исключения и построим модель при помощи метода К ближайших соседей. Поскольку расчеты могут занять продолжительное время, распараллелим вычисления. Для этого воспользуемся функциями из пакета **doParallel**. Создадим кластер из 10 ядер:

```
myCluster<-makeCluster(10, type="PSOCK")
registerDoParallel(myCluster)
```

Далее установим параметры перекрестного контроля для отбора признаков и подбора значений К:

```
trControl <- trainControl(method="cv",
                  number=10)

rfeControl <- rfeControl(functions = caretFuncs,
                  method = "cv",
                  number = 10,
                  verbose = FALSE)
```

Оценим точность прогноза на перекрестном контроле при использовании топ 100, 200, 300, 400, 500, 1000, 1500, 2000, 3000 лучших признаков с помощью функции **rfe**:

```
rfefit <- rfe(x=data.matrix(tr[,-1]), y=tr[,1],
          method="knn",
          preProcess=c("center", "scale"),
          sizes = c(100,200,300,400,500,1000,1500,2000,3000),
          rfeControl = rfeControl,
          trControl=trControl)
```

Наибольшая точность была получена при использовании топ 1000 лучших признаков и 5 ближайших соседей:

```
> rfefit

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables Accuracy  Kappa AccuracySD KappaSD Selected
       100   0.8125 0.5756     0.1301  0.2966
       200   0.8125 0.5756     0.1301  0.2966
       300   0.7982 0.5543     0.1486  0.3249
       400   0.8125 0.5756     0.1301  0.2966
       500   0.8125 0.5756     0.1301  0.2966
      1000   0.8149 0.5938     0.1244  0.2768        *
      1500   0.7982 0.5370     0.1486  0.3776
      2000   0.8006 0.5403     0.1439  0.3745
      3000   0.7982 0.5403     0.1325  0.2890
     10176   0.7696 0.4328     0.1762  0.4584

The top 5 variables (out of 1000):
   NCOA7, PLSCR1, SMCO4, UBE2E3, PRNP

> rfefit$fit

k-Nearest Neighbors

  70 samples
1000 predictors
   2 classes: 'N', 'Y'

Pre-processing: centered (1000), scaled (1000)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 64, 64, 63, 62, 63, 63, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.8428571  0.6381232
  7  0.8142857  0.5879954
  9  0.8267857  0.6094239

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

Оценим точность прогноза на тестовой выборке:

```
> pred<-predict(rfefit$fit,newdata=test.x)
> confusionMatrix(data=pred,reference=test.y,positive="ER(+)")

Confusion Matrix and Statistics

          Reference
Prediction  N  Y
         N  4  1
         Y  5 19

               Accuracy : 0.7931
                 95% CI : (0.6028, 0.9201)
    No Information Rate : 0.6897
    P-Value [Acc > NIR] : 0.1576

                  Kappa : 0.4494

 Mcnemar's Test P-Value : 0.2207

            Sensitivity : 0.9500
            Specificity : 0.4444
         Pos Pred Value : 0.7917
         Neg Pred Value : 0.8000
             Prevalence : 0.6897
         Detection Rate : 0.6552
   Detection Prevalence : 0.8276
      Balanced Accuracy : 0.6972

       'Positive' Class : Y
```

Из результатов прогноза следует, что точность прогноза составила 0.793.

Теперь отберем признаки при помощи метода имитации отжига:

```
trControl <- trainControl(method="cv", number=10)

safsControl <- safsControl(functions = caretSA,
                    method = "cv",
                    number = 10,
                    improve=30,
                    verbose = FALSE)

safsfit <- safs(x=tr.x, y=tr.y,
            method="knn",
            preProcess=c("center", "scale"),
            iters=100,
            safsControl = safsControl,
            trControl=trControl)
```

Наибольшая точность на перекрестном контроле была получена при использовании топ 2112 лучших признаков и 7 ближайших соседей:

```
> safsfit

Simulated Annealing Feature Selection

70 samples
10176 predictors
2 classes: 'ER(-)', 'ER(+)'

Maximum search iterations: 100
Restart after 30 iterations without improvement (2 restarts on
average)

Internal performance values: Accuracy, Kappa
Subset selection driven to maximize internal Accuracy

External performance values: Accuracy, Kappa
Best iteration chose by maximizing external Accuracy
External resampling method: Cross-Validated (10 fold)

During resampling:
  * the top 5 selected variables (out of a possible 10176):
    LOC100507507 (80%), ADAMDEC1 (70%), EGFL6 (70%), FKBP14 (70%),
GALNT14 (70%)
  * on average, 2291.1 variables were selected (min = 2064, max =
2507)

In the final search using the entire training set:
   * 2112 features selected at iteration 9 including:
     ADA, CDH2, C8orf88, ZNF667.AS1, NEBL.AS1 ...
   * external performance at this iteration is

   Accuracy       Kappa
     0.8071      0.5569

> safsfit$fit

k-Nearest Neighbors

  70 samples
2112 predictors
   2 classes: 'ER(-)', 'ER(+)'

Pre-processing: centered (2112), scaled (2112)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 63, 62, 64, 62, 63, 63, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.7136905  0.3475774
  7  0.7553571  0.4532335
  9  0.7327381  0.3479449

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 7.
```

Точность прогноза на тестовой выборке:

```
> pred<-predict(safsfit$fit,newdata=test.x)
> confusionMatrix(data=pred,reference=test.y,positive="ER(+)")
```

```
Confusion Matrix and Statistics

          Reference
Prediction ER(-) ER(+)
     ER(-)     3     1
     ER(+)     6    19

               Accuracy : 0.7586
                 95% CI : (0.5646, 0.897)
    No Information Rate : 0.6897
    P-Value [Acc > NIR] : 0.2796

                  Kappa : 0.3344

 Mcnemar's Test P-Value : 0.1306

            Sensitivity : 0.9500
            Specificity : 0.3333
         Pos Pred Value : 0.7600
         Neg Pred Value : 0.7500
             Prevalence : 0.6897
         Detection Rate : 0.6552
   Detection Prevalence : 0.8621
      Balanced Accuracy : 0.6417

       'Positive' Class : ER(+)
```

В конце работы не забудьте остановить кластер:

stopCluster(myCluster)

Практическое задание

В файле «Wilms tumor.txt» содержатся данные об уровне экспрессии генов в опухоли Вильмса. Часть образцов получена от пациентов, у которых впоследствии случился рецидив заболевания, а остальные получены от пациентов, у которых не случилось рецидива. Выполните отбор признаков при помощи метода фильтрации (тест Стьюдента) и метода рекурсивной элиминации признаков. Постройте классификационные модели для прогноза риска рецидива при помощи метода К ближайших соседей, метода опорных векторов, используя разные ядра, и метода Random Forest. Оцените точность прогноза на тестовой выборке. Какой метод наиболее точен?

Примечание: *поскольку объём данных небольшой, сгенерируйте несколько обучающих и тестовых выборок, а результаты прогноза усредните.*

В качестве самостоятельной работы (дома) постройте модель при помощи одного из методов, используя метод имитации отжига для отбора признаков.

Вопросы

1. Что представляет собой метод опорных векторов?
2. Что представляет собой метод рекурсивной элиминации признаков?
3. Что представляет собой метод имитации отжига?