Análisis predictivo en R (I)

SISTEMAS INTELIGENTES PARA LA GESTIÓN DE LA EMPRESA

CURSO 2016-2017

Construcción de predictores

Proceso

1. Crear el modelo utilizando la función de entrenamiento
   - **`fit <- train`**`(<formula>, <data>, <method>)`
2. Evaluar las propiedades del modelo
   - **`print, plot, summary`**`, …`
3. Predecir salidas para conjunto de instancias
   - **`predict(`***`fit`***`,`** `<samples>, [tipo de salida*]`**`)`**

`* valores de predicción, probabilidad, etc.`

caret

Interfaz unificada para creación y uso de >145 modelos de predicción

- Pfizer (2005)
- CRAN (2007)
- Web: http://caret.r-forge.r-project.org
- Información: http://www.jstatsoft.org/v28/i05/paper
- M. Kuhn, K. Johnson (2013) *Applied Predictive Modeling*. Springer.

caret

Creación del modelo

- Identificar variables de entrenamiento (exploración)
- Calcular los conjuntos de entrenamiento (*train*) y validación (*test*)
- Determinar los valores de los parámetros
- Lanzar algoritmo
- Medir rendimiento

  [repetir] + [preprocesamiento]

Objetivos

- Maximizar/minimizar la métrica de error (e.g.: %, *confusion matrix*, *ROC*, *logloss*)
  - En problemas no balanceados, sensitivity (TP) vs specificity (FP), AUC
- Evitar sobre-entrenamiento (*overfitting*)
  - K-fold cross-validation

`caret` – Pre-procesamiento

Pre-procesamiento: centrado, escalado y transformación

- **`preProcess`**
- Las transformaciones deben aplicarse a todos los datasets utilizados en el entrenamiento
  - Atención si se utiliza cross-validation

Evitar sobre-entrenamiento < Seleccionar adecuadamente los conjuntos de entrenamiento

- Seleccionar ejemplos aleatorios
  - **`createDataPartition`**
- Seleccionar ejemplos diferentes
  - **`maxDissim`**
- Crear grupos para *cross-validation*
  - **`createFolds`**
- Bootstrapping
  - **`createResample`**

`caret` – Ej.: Árboles de clasificación

Diversos paquete en R para clasificación con árboles

- **`rpart`**, **`RWeka`**, **`evtree`**, **`C50`**, **`party.rpart`**

Ejemplo: **`rpart`**

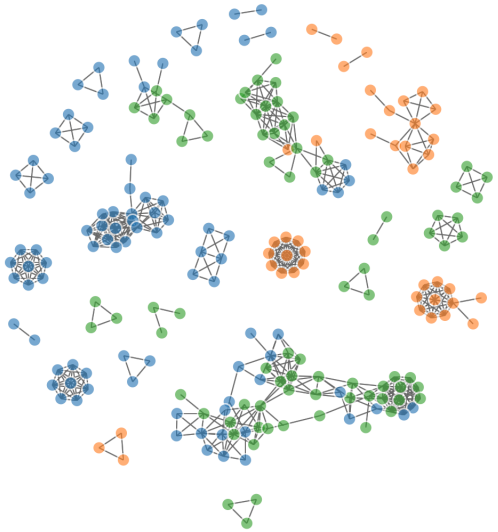
- maxdepth
  - Si no se especifica, utiliza como parámetro de parada C_p (cost-complexity parameter)
- Aplica podado del árbol para evitar *overfitting*
  - 10-fold CV + elige el árbol más sencillo con error en el entorno de un error estándar respecto al mejor árbol
- Emplea el % de error de clasificación como métrica de error

- ¿Ajustes del modelo de predicción?
  - 10-Fold CV puede introducir errores en conjuntos de datos pequeños
  - Modificar parámetros para permitir cierto *overfitting*
  - Utilizar otras métricas de error

`caret` – Métodos

Métodos disponibles

- Lista: http://topepo.github.io/caret/available-models.html
- Visualización: https://topepo.github.io/caret/models-clustered-by-tag-similarity.html



caret – Árboles de clasificación

```
train(Class ~ ., data = training, method = "rpart")

train(Class ~ ., data = training, method = "rpart", tuneLength = 30)


cvCtrl <- trainControl(method = "repeatedcv", repeats = 3)
train(Class ~ ., data = training, method = "rpart",
      tuneLength = 30,
      trControl = cvCtrl)


cvCtrl <- trainControl(method = "repeatedcv", repeats = 3,
                       summaryFunction = twoClassSummary,
                       classProbs = TRUE)
set.seed(1)
rpartTune <- train(Class ~ ., data = training, method = "rpart",
                   tuneLength = 30,
                   metric = "ROC",
                   trControl = cvCtrl)
```

`caret` – Objeto `train`

`plot.train`

- Visualizar gráfico del modelo de predicción con diferentes ajustes de configuración

`print.train`

- Descripción en formato texto de los resultados

`predict.train`

- Predecir nuevos ejemplos (por ejemplo, datos de test)

`$finalModel`

- Visualizar gráfico del modelo de predicción con diferentes ajustes de configuración

`caret` – Datos no balanceados

http://topepo.github.io/caret/subsampling-for-class-imbalances.html

Sub-sampling (!= *split*, normalmente integrado dentro del *resampling*)

- *Down-sampling:* Reducir número de ejemplos de la clase mayoritaria **`downSample`**
- *Up-sampling*: Seleccionar ejemplos de la clase minoritaria con reemplazo **`upSample`**
- Híbrido: Crear nuevos ejemplos sintéticos **`DMwR`**, **`ROSE`**

Cost-sensitive learning, Ensembles

- Especificar matriz de costes para las clases
- Utilizar un método que acepte entrenamiento con costes
  - Cost-sensitive C5.0
  - Cost-sensitive CART