quantil

Applied Mathematics

Random Forests

Abril 2014

Felipe Parra

Por que Arboles para Clasificación

- Interpretación intuitiva
- Pueden ajustarse a relaciones muy complejas entre los predictores y la respuesta
- Selección de variables de manera automática
- Manejan de manera muy natural predictores y respuestas categóricas
- No hay que preocuparse por transformar los predictores
- Bajo costo computacional (en comparación con otros métodos)

Como funcionan

- En este caso la devianza no se pude definir como el RSS
- Se define en este caso el cross-entropy

$$\text{cross-entropy} = -2 \times \sum_{i=1}^{n} \sum_{k=0}^{K-1} \widehat{p}_{ik} \log \widehat{p}_{ik}$$

- Por convención se define log(0)x0=0
- A medida que una probabilidad individual tiende a uno o cero el sumando tiende a cero
  - Si el modelo tiene mayor confianza en su predicción el cross-entropy tiende a cero
- Si una probabilidad tiende a 1/K (el modelo esta perdido) en este caso se esta maximizando la devianza

Como funcionan

- Para cada nodo la predicción final va a estar determinada por la cantidad de observaciones de cada tipo que caen en el nodo

- Ejemplo:

	0	1	2	3
1	0.0000000	0.5714286	0.0000000	0.4285714
2	0.1000000	0.6000000	0.3000000	0.0000000
3	0.2857143	0.3571429	0.3571429	0.0000000
4	0.0000000	0.6000000	0.2000000	0.2000000
5	0.0000000	0.0000000	0.4000000	0.6000000
...				

Como seleccionar el mejor árbol

- Paso 1: 'Greedy Algorithm'
  - Crecer el árbol al máximo, creando una nueva rama a la vez
  - En cada paso se pregunta que partición del árbol genera la mínima devianza
  - Cada partición (decisión) involucra un solo predictor

Como seleccionar el mejor árbol

- Paso 2: 'Podar el árbol'
  - El árbol del paso 1 es podado buscando el modelo optimo
  - Podría imaginarse como una búsqueda extensiva sobre todos los posibles sub-arboles tratando de minimizar una combinación de devianza y complejidad:

$$\text{deviance} + \alpha \times \text{number of terminal nodes in tree}$$

  - Alpha en este caso es el parámetro de suavizamiento
    - Mayor alpha implica menor complejidad en nuestro árbol optimo)\
  - Alpha es seleccionado usando 'K-Fold' Cross Validation
    - Para cada alpha se intenta minimzar la devianza

Ahora si, Random Forests

- Utilizan la tecnica de Bagging (Bootstrap Aggregation)
  - Promediando varios estimadores 'no tan buenos' se obtiene un muy buen estimador
- Considerados el mejor clasificador 'Off the shelf'
- El algoritmo:
  - Para b=1,2,…,B
    - Tomar una muestra con remplazo de 'n' datos
    - Crecer el árbol con la muestra de bootstrap el máximo posible bajo la condición de que cada nodo debe contener por lo menos 's' observaciones
      - **La salsa secreta**: escoger tan solo 'm' de los 'p' predictores para tomar la decisión en cada partición de una nueva rama aleatoriamente.
  - Guardar los B arboles del bootstrap

Algunos comentarios

- Para clasificación: democrático
- Para regresión: promedio

- Por default en R:
  - B=500
  - m = parte entera de p/3 para regresión y raíz de p para clasificación

- **Lo mejor**: no hay necesidad de hacer cross validation para escoger los parametros optimos:
  - Usar el 'out-of'bag' (OOB) sample para probar las capacidades predictivas del random forest para cada conjunto de parametros

Random Forests vs Boosting

- Boosting parece dominar al bagging en la mayoria de las aplicaciones
- La idea del bagging es promediar estimadores ruidosos insesgados para reducir la varianza
  - Ideal en situaciones de varianzas grandes y estimadores insesgados como los arboles
  - Random forests construye arboles decorrelacionados (Produce resultados similares a boosting pero son mas fáciles de entrenar)
- Los random forests reducen la correlación entre los arboles, sin incrementar mucho la varianza, al escoger las variable de manera aleatoria al crecer el arbol.
  - La correlacion entre los arboles es de 0.05 en general

$$\frac{1}{B}\sigma^2 \qquad \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

…ahora si lo mejor

- Muy Fácil de implementar en R!
- Paquete randomForest
- Solo se necesitan dos comandos
  - randomForest()
  - tuneRF()
    - Escoge m que es el parámetro mas importante

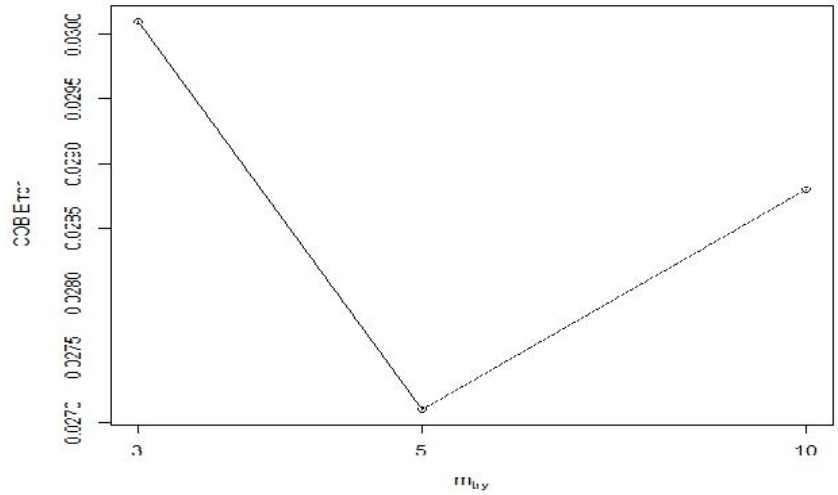
Aplicación

- Competencia de HFT:
  - http://www.circulumvite.com/home/trading-competition
- Un proxy del USD/BRL (multiplicado por 2000) tomando los mejores BID y ASK
- 27 Predictores: Indicadores de mercado
  - Tendencia (suavizamientos)
  - Diferencias en tamaños de BID y ASK
  - Etc., (todos propietary indicators)
- El objetivo: Clasificar si la tasa de cambio va a aumentar en los próximos 120 milisegundos

Modelo 1

- Tomar aleatoriamente 10,000 observaciones como muestra de entrenamiento
- Entrenar el random forest (escoger el mejor ‘m’)

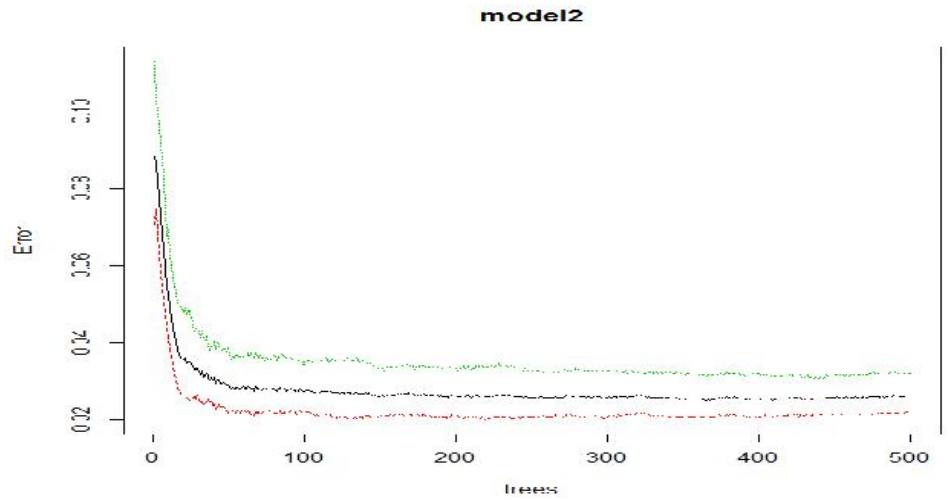
```
xmat = trainset[,1:27]
tuneRF(xmat,factor(trainset$change))
```



Modelo 1

- Correr el random forest con el ‘m’ optimo

```
model2 = randomForest(formula = factor(change) ~ ., data = trainset, mtry=5)
plot(model2)
table(testset$change, predict(model2,testset))
```



- 96.4% de las subidas en la tasa de cambio clasificadas de manera correcta!

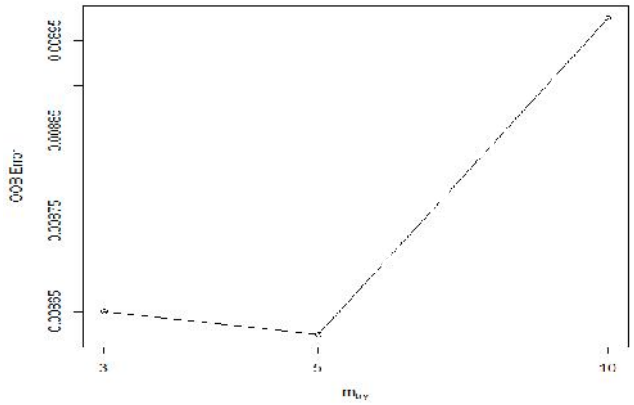
```
      0     1
0 24006   484
1   681 18181
```

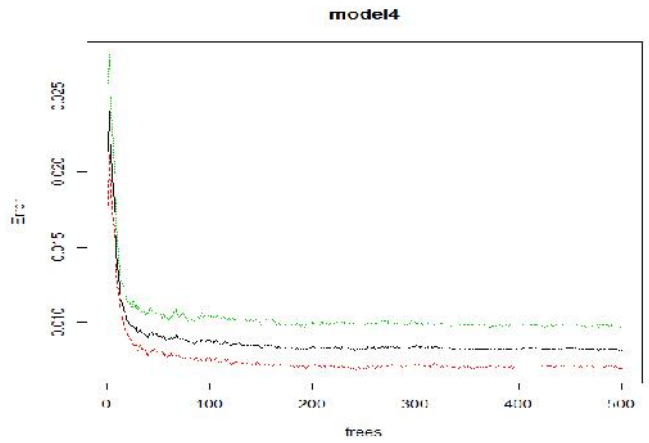
Sospechoso no?!

- Donde esta ‘la trampa’
  - Tomar la muestra de entrenamiento de manera aleatoria….

Modelo 2

- Se entreno el random forest con las primeras 40,000 observaciones





- Esta vez el error es del 48.6% !!!

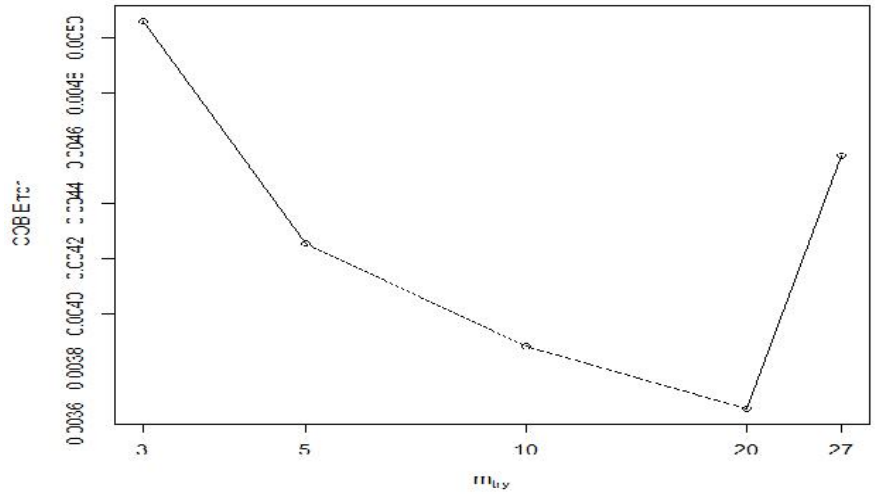
```
     0    1
0 4308 3142
1 2867 3035
```

Apague y vamonos?

- ….Este Parrita

Modelo 3

- Se entrenó el random forest con todas las observaciones de un dia y se aplicó al dia siguiente



- M=20 suena como grande…riesgo de overfitting

Modelo 3

- Probar m=3, 5, 10 para ver cual tiene el mejor desempeño fuera de muestra
  - **m=10 (38.5% de las subidas bien clasificadas)**

	0	1
0	16253	10573
1	16250	10169

  - **m=5 (59.9% de las subidas bien clasificadas)**

	0	1
0	10594	16232
1	10585	15834

  - **m=3 (60.3% de las subidas bien clasificadas)**

	0	1
0	10594	16232
1	10585	15834

Conclusiones

- No es la receta de la alquimia…pero
- Tener cuidado con el overfitting
- Muy facil de implementar en R!

Bibliografía

- Notas de Clase de 'Statistical and Machine Learning Methods for Financial Data'. Chad Schafer. CMU. 2014

- The Elements of Statistical Learning, (2009). Hastie, Tibshirani & Friedman. Springer.