Bagging y Random Forests

Jorge Gallego

Facultad de Economía, Universidad del Rosario

marzo 14 de 2017

Introducción

- Vimos que los árboles de decisión son muy poderosos
- Sirven para hacer predicciones numéricas o categóricas
- No requieren supuestos distribucionales sobre los datos
- Son aplicables en diferentes contextos
- Además, son fáciles de explicar sin necesidad de recurrir a un lenguaje matemático

Introducción

- Sin embargo, no están exentos de debilidades
- Es fácil sobreestimar o subestimar el modelo
- Lo cual lleva a que en ocasiones la precisión alcanzada con datos fuera de la muestra no sea alta
- Lo vimos con el ejemplo del default crediticio. La precisión no era tan alta
- También tienen problemas de varianza de entrenamiento alta

Introducción

- Pequeños cambios en los datos de entrenamiento pueden llevar a grandes cambios en las estimaciones
- Los árboles pueden ser muy distintos, así los datos vengan de la misma población
- Hoy veremos dos técnicas que se basan en los árboles de decisión
- Pero que resuelven algunos de estos problemas, logrando menor varianza y estimaciones más precisas
- En particular, veremos *bagging* y *random forests*

Bootstrapping

- Para entender la lógica detrás del *bagging* debemos entender el procedimiento de *bootstrapping*
- En estadística, consiste en hacer un test o métrica con base en muestreo aleatorio con reemplazo
- La idea es obtener muestras aleatorias de los datos, para calcular a partir de esas muestras algún estadístico
- Por ejemplo, la desviación estándar. Es muy útil en situaciones en las que es difícil o imposible hacerlo directamente

Bagging

- En este contexto, usaremos bootstrapping para mejorar métodos de aprendizaje estadístico como los árboles
- Como dijimos antes, los árboles de decisión tienen problemas de alta varianza
- Supongamos que dividimos aleatoriamente los datos de entrenamiento en dos partes iguales
- Si estimamos dos árboles de decisión para cada mitad, es posible que obtengamos resultados muy distintos
- El *bagging*, o agregración boostrap, es un procedimiento para reducir la varianza de un método de aprendizaje estadístico

Bagging

- Es particulamente útil en el contexto de los árboles de decisión
- Supongamos que tenemos un conjunto n de observaciones independientes $Z_1, \ldots, Z_n$
- Cada una con varianza σ^2
- Si las promediamos, la varianza de la media de las observaciones, $\bar{Z}$, es σ/n
- Luego, promediar las observaciones sirve para reducir la varianza

Bagging

- Esto lo podemos aplicar al contexto de los modelos de aprendizaje automatizado
- Tomamos muchos conjuntos de entrenamiento de la población
- Construimos un modelo predictivo con cada conjunto
- Y promediamos las predicciones de cada modelo para llegar a una predicción definitiva
- Esto nos ayuda a alcanzar predicciones más precisas

Bagging

- Llamemos $\hat{f}^b(x)$ a la predicción que hacemos del ejemplo x con el modelo $\hat{f}(\cdot)$ entrenado con los datos b
- Por ejemplo, la predicción que obtuvimos de un árbol de decisión
- Si calculamos $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^B(x)$ usando B datos de entrenamiento distintos,
- Podemos obtener un único modelo de aprendizaje automatizado de baja varianza al promediar:

$$\hat{f}_{prom}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x)$$

Bagging

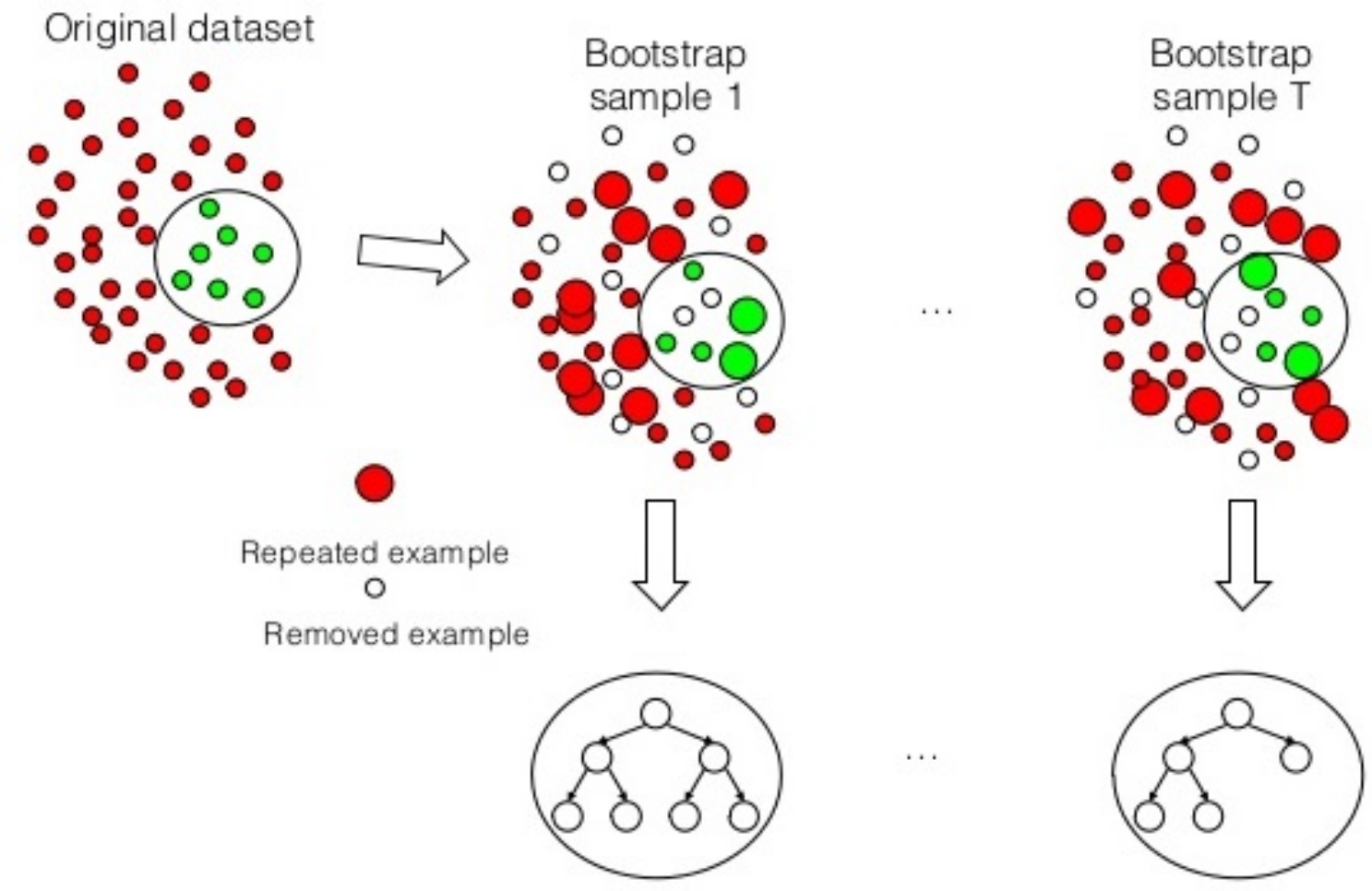
- Pero esto en la práctica no es posible o útil porque no tenemos múltiples datos de entrenamiento
- En su lugar, podemos hacer bootstrap
- Obtener muestras repetidas de un único conjunto de datos de entrenamiento
- Entrenamos el modelo en cada uno de los b conjuntos de datos de entrenamiento para obtener $\hat{f}^{*b}(x)$:

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x)$$

Bagging

- En eso consiste el bagging. ¿Cómo lo aplicamos a los árboles de decisión?
- Construimos B árboles de decisión usando B datos de entrenamiento de bootstrap
- Pero cuando hacemos clasificación nuestros outcomes no son cuantitativos, sino categóricos
- En lugar de promediar, usamos regla de la mayoría (cada árbol "vota")
- Así, la predicción final para un ejemplo es la que hagan la mayoría de árboles

Bagging



Fuente: https://www.slideshare.net/mlvlc/l4-ensembles-of-decision-trrees

Bagging

- ¿Cuántos árboles construir?
- Afortunadamente, el parámetro B no es crucial en este método
- Usar muchos árboles no lleva a sobreestimar el modelo
- Los resultados se estabilizan después de cierto número de árboles

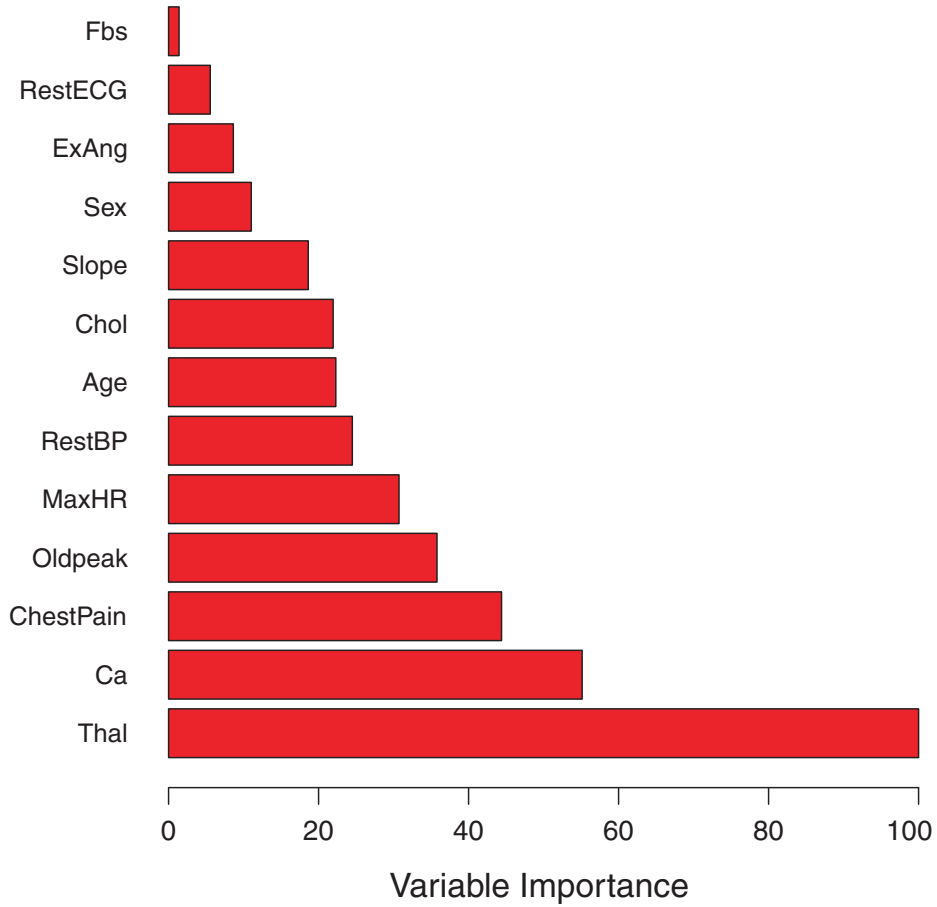
Bagging

- Al hacer bagging reducimos el problema de alta varianza y aumentamos la precisión
- Pero esto tiene un costo: el modelo ya no es intuitivo ni fácil de interpretar
- Si empaquetamos varios árboles ya no los podemos representar por medio de uno solo
- Tampoco es claro cuáles son las variables más importantes al momento de clasificar

Bagging

- Sin embargo, sí podemos obtener un resumen de la importancia de cada predictor
- En clasificación, utilizamos el índice de entropía o Gini que hayamos usado para calcular la ganancia de información
- Sumamos la cantidad total en la que se reduce el índice al hacer divisiones en cada predictor
- Y promediamos en los B árboles. Los predictores que promedien más alto son los más importantes

Ejemplo: Prediciendo Enfermedad de Corazón



Fuente: James et al. (2013)

Error *Out of Bag*

- Hay una forma muy interesante de evaluar la precisión de este tipo de modelos
- Cuando obtenemos una muestra bootstrap, para construir un árbol, hay observaciones, que se quedan por fuera
- Así, cada observación tiene un conjunto de árboles que no la "utilizaron"
- Esos árboles pueden hacer una predicción sobre esa observación
- Para luego llegar a una predicción agregada sobre la observación

Error *Out of Bag*

- Lo interesante es que podemos comparar, para cada observación, la predicción *Out of Bag* (OOB)
- Con el verdadero estado de la observación
- Lo cual nos da una tasa de error OOB del modelo
- Naturalmente, nos interesa minimizar este error
- Como también el error tradicional, cuando predecimos en la base de entrenamiento

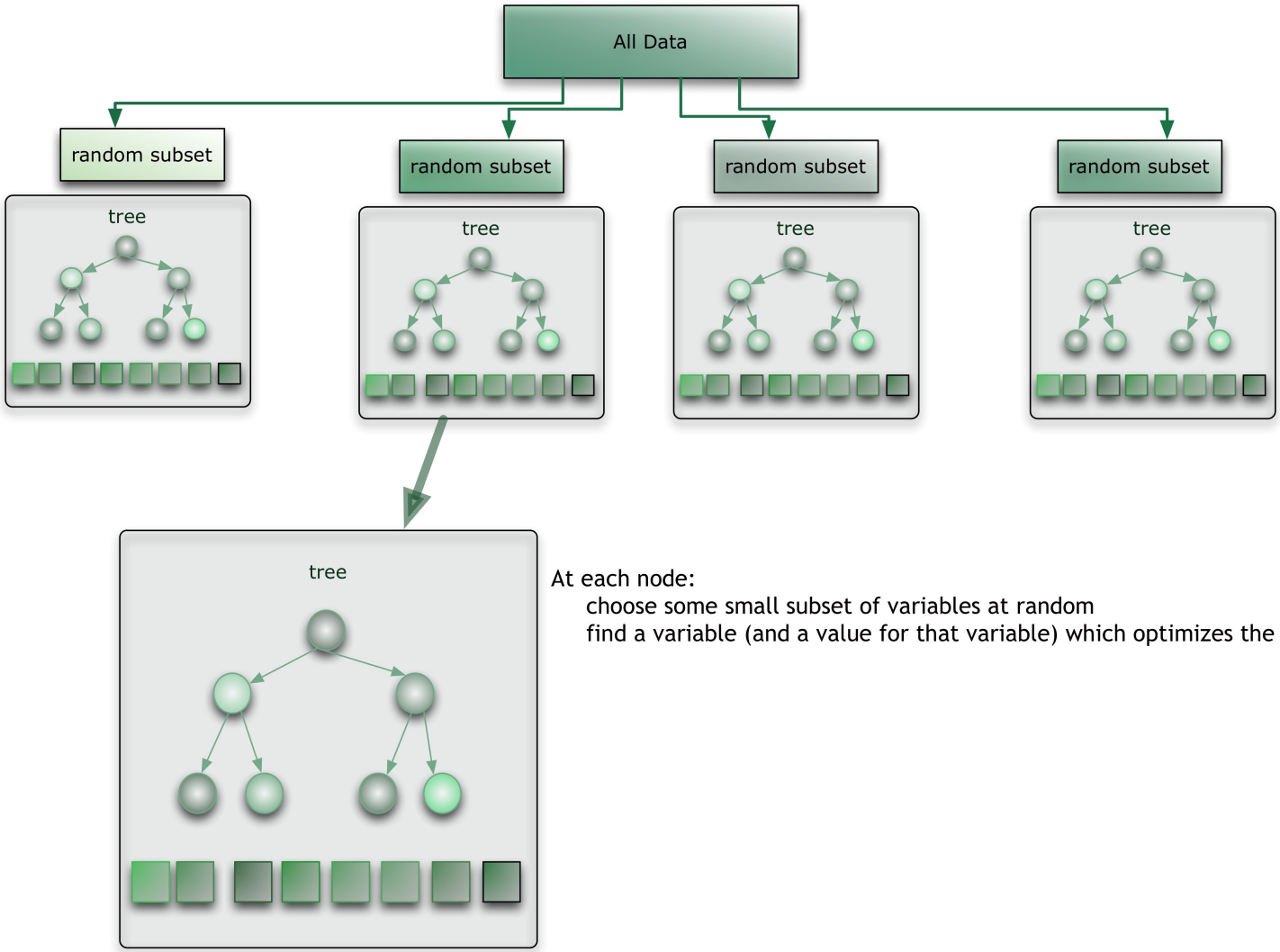
Random Forests

- Los *random forests* son una mejora respecto al *bagging*
- En bagging los árboles se construyen respecto a muestras aleatorias
- Pero cada árbol usa exactamente el mismo conjunto de características (variables)
- Luego cada árbol debe usar las mismas características para predecir
- Quizás en distintos órdenes y con diferentes valores para hacer los *splits*

Random Forests

- Por ende, los árboles tienden a estar muy correlacionados
- Si hay regiones del espacio de características donde un árbol tiende a cometer errores, todos los árboles tienden a hacerlo
- Los *random forests* buscan corregir este asunto, disminuyendo la correlación entre árboles
- Lo hacen aleatorizando el conjunto de variables que cada árbol puede utilizar

Random Forests



At each node:
choose some small subset of variables at random
find a variable (and a value for that variable) which optimizes the split

Fuente: http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics

Random Forests

Para cada árbol individual del bosque, el método hace lo siguiente:

1. Extrae una muestra *bootstrap* de los datos de entrenamiento

2. Para cada muestra, construye un árbol de decisión. En cada nodo del árbol:

    i) Selecciona aleatoriamente un subconjunto de variables del total de características disponibles
    ii) Selecciona la mejor variable y el mejor *split* de ese subconjunto de características
    iii) Continua hasta que el árbol crece por completo

3. El conjunto final de árboles se empaqueta y se hace la predicción

Random Forests

- ¿Cuántas variables seleccionar en cada nodo?
- La heurística es tomar $m = \sqrt{p}$, donde p es el número total de variables
- Afortunadamente, los resultados no son muy sensibles a la cantidad de variables seleccionadas
- Valores menores hacen que los árboles crezcan más rápido
- Pero si hay muchas variables pero solo pocas son útiles, seleccionar más aumenta la prob. de escoger las que sirven

Random Forests

- ¿Cuál es la lógica de usar solo un subconjunto de variables?
- Supongamos que hay una característica que es muy importante, frente a otras de importancia moderada
- Entonces, en el bosque la mayoría de los árboles empezarán haciendo *split* en dicha característica
- Luego, los árboles serán parecidos entre sí. Estarán muy correlacionados
- El bosque será muy homogéneo. Promediar en cantidades homogéneas no ayuda a reducir la varianza

Random Forests

- Los *random forests* resuelven este problema
- No todos los árboles incluirán al predictor importante
- Así, los árboles se parecerán menos entre sí
- Todo lo cual disminuye la correlación entre ellos y reduce la varianza
- Nótese que *bagging* es un caso particular de *random forests*: cuando $m = p$