Métodos de Clasificación Basados en Árboles (II)

EP7144 - Técnicas de Minería de Datos

Mag. Enver Gerald Tarazona Vargas
etarazona@lamolina.edu.pe
envertv@gmail.com

Escuela de Post-Grado

Universidad Nacional Agraria La Molina (UNALM)

Maestría en Estadística Aplicada

Comentarios Generales I

- Bootstrap es usado en muchas situaciones en las cuales es complicado o imposible calcular directamente la desviación estándar de una cantidad de interés.
- Los árboles de decisión padecen por tener una gran varianza.
- Esto quiere decir que si nosotros dividimos la data de entrenamiento en dos partes de manera aleatoria y ajustamos un árbol de decisión en ambas mitades, los resultados pueden ser bastante diferentes.
- *Bootstrap aggregation*, or, *bagging*, es un procedimiento que busca reducir la varianza en un método de aprendizaje estadístico.

Comentarios Generales II

- Recordar: Dada un conjunto de observaciones independientes $Z_1, \ldots, Z_n$, cada uno con varianza σ^2, la varianza de la media $\overline{Z}$ de las observaciones esta dada por $\frac{\sigma^2}{n}$.
- En otras palabras, el promediar un conjunto de observaciones reduce la varianza.
- Una forma natural de reducir la varianza e incrementar la precisión en la predicción de un método de aprendizaje estadístico es tomar muchas muestras de entrenamiento de la población, construir un modelo predictivo por separado usando cada conjunto de entrenamiento y promediar los resultados de las predicciones..

Comentarios Generales III

- En otras palabras, podemos calcular $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^B(x)$ usando B conjuntos de entrenamiento separados, y promediarlos para obtener un modelo de aprendizaje estadístico de baja varianza dado por:

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

- Obviamente este enfoque no es práctico porque generalmente no tenemos acceso a múltiples conjuntos de entrenamiento.
- De manera alternativa, podemos aplicar bootstrap, tomando muestras repetidas de un sólo conjunto de entrenamiento.
- Bajo este enfoque generamos B diferentes muestras de entrenamiento por bootstraping.

Comentarios Generales IV

- Se entrena el método en la b-ésima muestra de entrenamiento para obtener $\hat{f}^{*b}(x)$ y finalmente promediar todas las predicciones obteniendo:

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

  El cual es llamado bagging

- El método anterior es utilizado en árboles de regresión. Para el caso de clasificación, se registra la clase predicha para cada observación en cada uno de los B árboles y se toma aquella con mayor frecuencia (voto mayoritario).

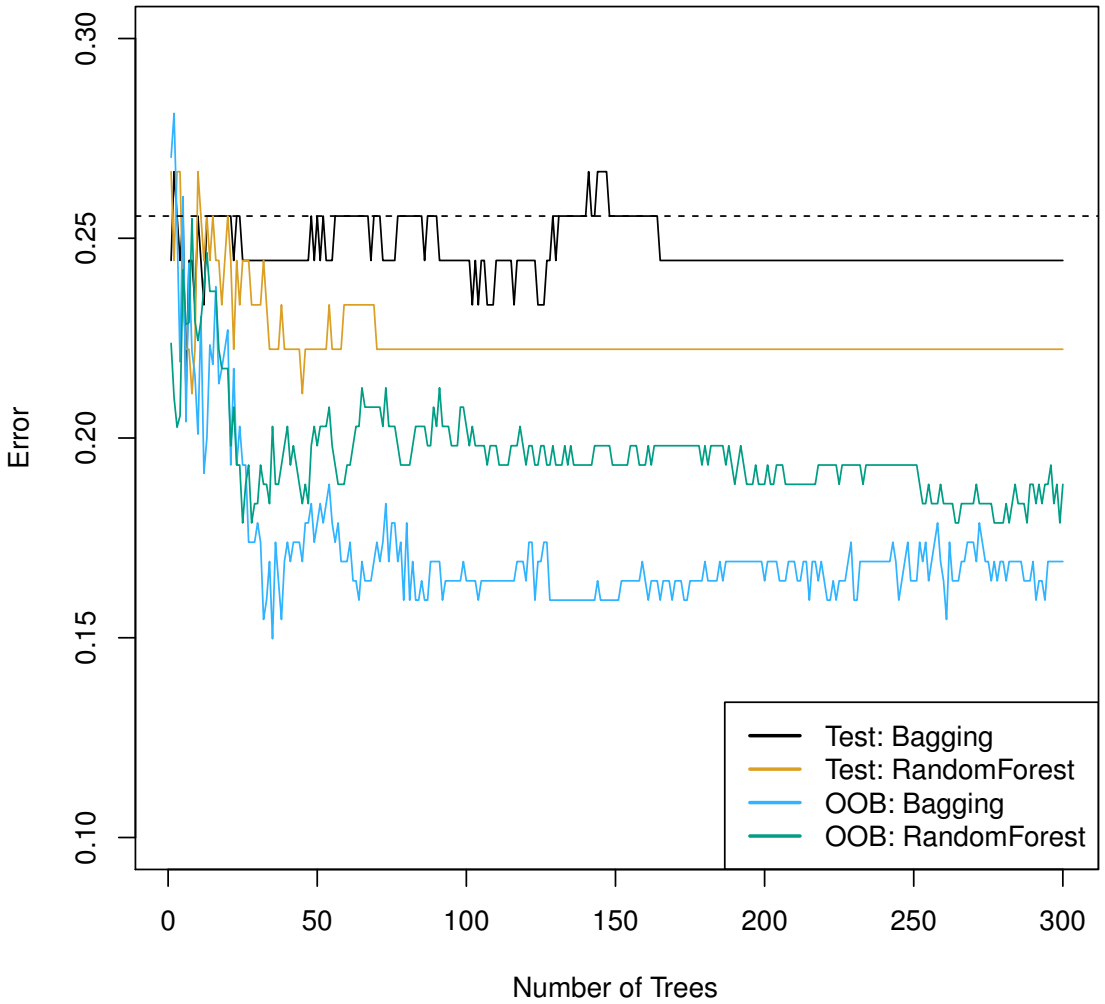
Error de Estimación *Out-of-Bag* I

- Hay una forma sencilla de estimar el error de estimación de un modelo agregado por bagging sin la necesidad de usar validación cruzada o usar el enfoque de usar una muestra de validación.
- Es posible demostrar que, en promedio, cada árbol agregado por bagging usa en promedio alrededor de dos tercios de las observaciones (ver James et al, capítulo 5).
- El tercio restante de las observaciones que no son usada para ajustar el árbol son conocidas como las observaciones out-of-bag (OOB).
- Podemos predecir la respuesta para la i-ésima observación usando cada uno de los árboles en que dicha observación estaba OOB.

Error de Estimación *Out-of-Bag* II

- Esto da alrededor de $B/3$ predicciones para la i-ésima observación.
- Para obtener una predicción individual de la i-ésima observación se usa la estrategia de voto mayoritario.
- La predicción OOB puede ser obtenida de esta forma para cada una de las n observaciones, para los cuales se calcula el error de clasificación.
- El error OOB resultante es una estimación válida del error de prueba para el modelo agregado por bagging, dado que la respuesta para cada observación es predicha usando solamente los árboles en los cuales la observación no fue usada para el ajuste.

Gráfica de Error en la clasificación



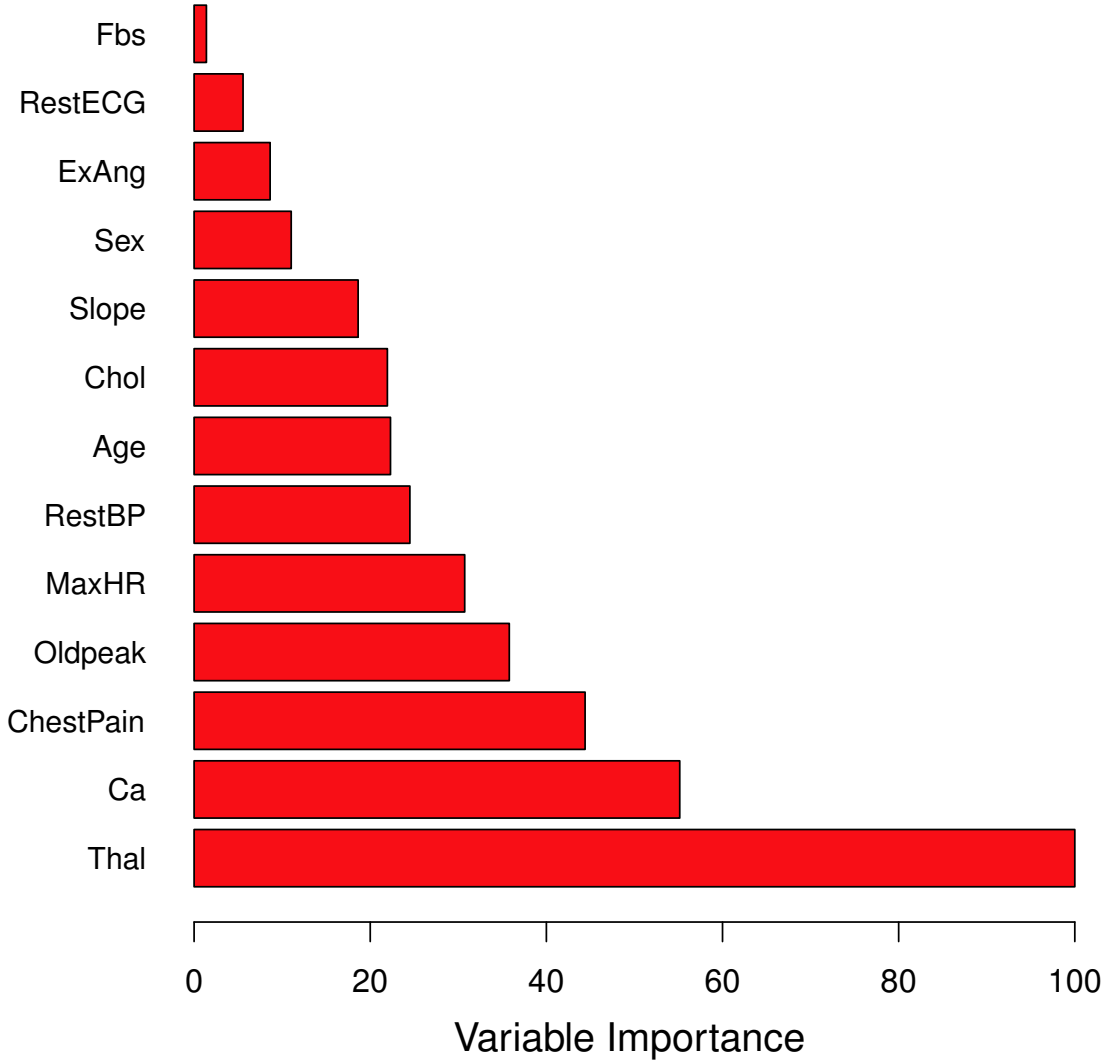
Medidas de la Importancia de las Variables I

- Bagging típicamente resulta en una mejora de la precisión sobre la predicción usando un sólo árbol.
- Desafortunadamente, puede ser difícil de interpretar el modelo resultante.
- Recordar que una de las ventajas de los árboles de decisión era la facilidad para interpertar el diagrama resultante.
- Cuando se agregan un gran número de árboles, no es posible representar el procedimiento estadístico de aprendizaje usando un sólo árbol, y no queda tan claro cuales son la variables más importantes para el procedimiento.
- En otras palabras, bagging mejora la precisión en la predicción a expensas de la interpretabilidad.

Medidas de la Importancia de las Variables II

- A pesar de que la colección de árboles agregados es más difícil de interpretar que un sólo árbol, es posible obtener un resumen de la importancia de cada predictor usando el índice de Gini.
- El resumen puede obtenerse añadiendo la cantidad total en que el índice de Gini es reducido por las divisiones sobre el predictor, promediado sobre todos los B árboles.

Gráfica de Importancia de las Variables



Random Forest I

- Random forests proporcionan una mejora sobre los árboles agregados por bagging realizando un pequeño ajuste que descorrelaciona a los árboles.
- De manera similar al bagging, se construye un número de árboles de decisión realizando un bootstrapping de las muestras de entrenamiento.
- Cuando se construye el árbol de decisión, cada vez que se considera una división, una muestra aleatoria de m predictores es seleccionada como candidatos para la división dentro del conjunto total de p predictores.
- La división es permitida de usar solamente uno de los m predictores.

Random Forest II

- Una nueva muestra de m predictores es tomada en cada división, y típicamente se elige $m \approx \sqrt{p}$.
- En otras palabras, en cada división del árbol, el algoritmo no permite que se consideren a la mayoría de los predictores.
- Si existiera un predictor demasiado fuerte en el conjunto de datos, mientras el resto de los predictores tiene una fuerza moderada, entonces la mayoría de árboles agregados por bagging usarán este predictor al inicio de la división.
- Como consecuencia la mayoría de los árboles lucirán similares unos a los otros y las predicciones estarán altamente correlacionadas.
- Desafortunadamente, promediar muchas cantidades altamente correlacionadas no ayudan a conseguir una gran reducción de la varianza.

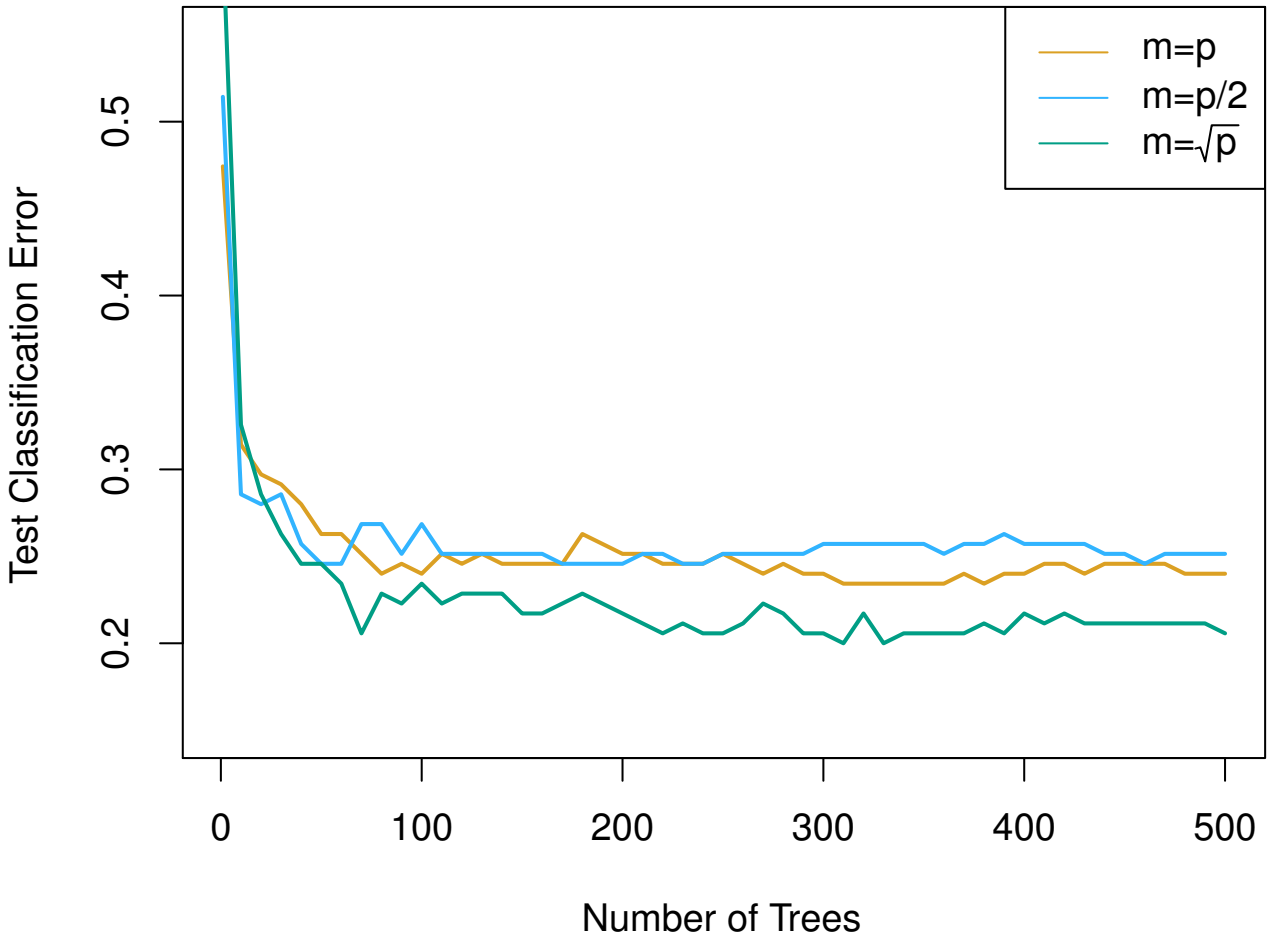
Random Forest III

- en otras palabras, bagging no logrará una substancial reducción de la varianza en comparación con un sólo árbol en situaciones de este tipo.
- Random forests supera este problema forzando a que cada partición considere solamente un subconjunto de predictores.
- En promedio, $(p - m)/p$ de la divisiones no considerarán el predictor más fuerte, de tal forma que otros predictores tienen la oportunidad de ser seleccionados.
- Podemos pensar en este proceso como una descorrelación de los árboles, haciendo que el promedio de los árboles sea menos variable y más fiables.
- La principal diferencia entre bagging y random forests es la elección del subconjunto de predictores m.

Random Forest IV

- Si random forest es construido con $m = p$ es similar a realizar bagging.
- Usar un valor pequeño de m en la construción de random forest será de gran ayuda cuando se tiene una gran cantidad de predictores correlacionados.

Resultados con Random Forests



Boosting y AdaBoost I

- Boosting y la variante más usada AdaBoost (Adapting Boosting) genera un conjunto de clasificadores.
- Sin embargo, Adaboost los genera secuencialmente (Bagging los puede generar en paralelo).
- A todos los ejemplos, les asigna inicialmente un peso igual $(1/m)$.
- Cada vez que se genera un clasificador, se cambian los pesos de los nuevos ejemplos usados para el siguiente clasificador.
- La idea es forzar al nuevo clasificador a minimizar el error esperado. Para esto se les asigna más peso a los ejemplos mal clasificados y menos a los bien clasificados.

Boosting y AdaBoost II

- La idea es alentar crear modelos que se vuelvan "expertos" en los datos que no pudieron ser explicados por los modelos anteriores.
- Después de cada interacción los pesos reflejan que tan seguido las instancias han sido mal clasificadas por los clasificadores que se tienen hasta ese momento.
- Se generan igual B clasificadores de muestras de ejemplos pesadas. El clasificador final se forma usando un esquema de votación pesado que depende del desempeño de cada clasificador en su conjunto de entrenamiento.
- Este algoritmo requiere de clasificadores débiles que cambian su estructura con cambios en los datos y que no dan errores mayores al 50%.

Boosting y AdaBoost III

- El algoritmo se para cuando el error en los datos de entrenamiento pesados son mayores o iguales a 0.5 o cuando el error es cero (donde todos los pesos de las instancias se vuelven 0).
- Si no se pueden incorporar ejemplos pesados dentro del clasificador, se puede tener un efecto parecido por medio de un muestreo con reemplazo, seleccionando los ejemplos de acuerdo a su peso.
- Se puede tener problemas de underflow, por lo que es común eliminar ejemplos con pesos muy pequeños.
- Bagging sin pruning a veces reduce el error, Boosting sólo lo aumenta.

Boosting y AdaBoost IV

- En Boosting, si un clasificador tiene error cero, recibe recompensa infinita y es el unico ganador, por lo que generalmente se elimina.