

Aprendizado em conjunto

Carlos Diego Rodrigues

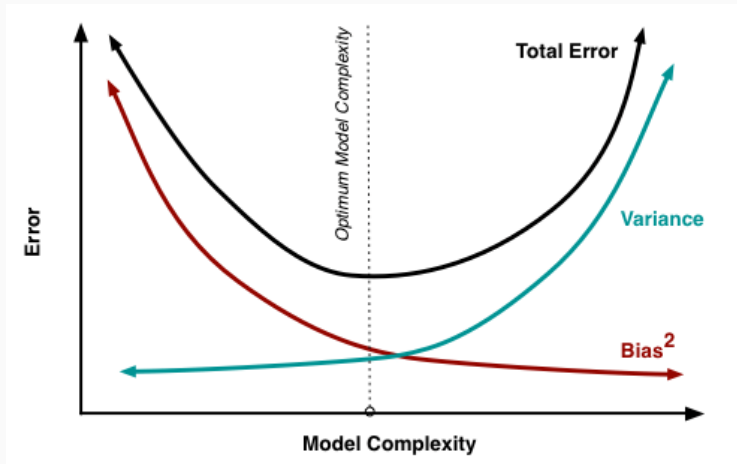
11 de janeiro de 2022

Universidade Federal do Ceará

Aprendizado em conjunto

- Como combinar diversos classificadores para gerar um classificador mais forte?
 - Classificadores têm limites!
 - Diferentes algoritmos podem dar diferentes resultados.
 - Diferentes conjuntos de treinamento podem dar diferentes resultados.
- Dois tipos:
 - Homogêneo
 - Heterogêneo
- Três formatos de combinação:
 - *Bagging*
 - *Boosting*
 - *Stacking*

Complexidade x Erro



Bagging

- Recomendado para modelos fracos e homogêneos com alta variância.
- Consiste na execução de um mesmo modelo para diversos conjuntos de entrada diferentes.
- Possivelmente utilizando *bootstrap*: amostragem do conjunto de dados original para criar novos conjuntos de entrada com possíveis replicações de instâncias.
- O resultado é algum tipo de média dos modelos de entrada, reduzindo portanto a variância dos modelos originais.
- No caso do problema de classificação uma votação entre os modelos originais decide qual é a classe de uma nova instância.
- Pode-se considerar (caso o método original permita) uma distribuição de probabilidade entre as classes que pode ser combinada através de uma função de verossimilhança.

Bootstrap

Atributos				Classe
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Aleatorização dos algoritmos

- Alguns métodos assimilam o processo de aleatorização dentro do próprio funcionamento.
- Neste caso sementes aleatórias diferentes sobre um mesmo conjunto de dados podem dar diferentes respostas.
- Caso o resultado incorra em grande variância, pode-se utilizar o *bagging* para a redução dessa variância.
- Um algoritmo popular para exemplificar este caso é o **random forests**.
- Uma das grandes vantagens do *bagging* é a paralelização da execução dos modelos fracos.

Boosting

- Assim como o *bagging*, o método de *boosting* é aplicado a um conjunto de modelos fracos e homogêneos.
- ao contrário do *bagging*, porém, a execução dos modelos é sequencial.

Atributos				Classe
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

Funcionamento do *boosting*

- A cada exemplo é associada um fator de relevância, inicialmente igual para todos.
- Então ao final da execução de um modelos os fatores de relevância são atualizados de forma que os exemplos que não foram classificados corretamente se tornam mais relevantes.
- A ideia é que os modelos que virão a seguir darão uma prioridade maior aos exemplos mais relevantes na sua avaliação para construção das regras.
- Isto também contribui para que o modelo composto (em conjunto) tenha um viés menor, já que um ajuste sobre a importância de cada exemplo (e conseqüentemente suas características) é feito ao longo do processo.
- Um famoso exemplo de algoritmo de *boosting* é o **AdaBoost**, juntamente com o *Gradient Boosting* e o *XGBoost*.

Características do *boosting*

- Assim como no *bagging*, modelos homogêneos são utilizados para formar um modelo melhor.
- Preferência para modelos simples, com alto viés.
- A sequência com que os modelos são escolhidos influencia no resultado dos demais pela calibragem da relevância das instâncias.
- Para o processo de classificação, um método de agregação dos resultados dos modelos fracos de acordo com a relevância do modelo é executado.
- Não pode ser paralelizado, mas os resultados obtidos são bastante relevantes.

- Neste formato de combinação de classificadores podemos considerar algoritmos distintos como modelos parciais, ou seja, uma combinação heterogênea.
- Vamos utilizar um dos classificadores que receberá o resultado obtido pelos demais para fazer a predição final sobre um determinado exemplo.
- Este classificador final é chamado **metaclassificador**.
- Vamos dividir o conjunto de dados de treinamento em dois: um conjunto de treinamento dos classificadores e o conjunto de treinamento do metaclassificador.

Stacking

- Em uma etapa inicial vamos realizar o treinamento dos classificadores parciais (ou fracos).
- Em seguida, vamos utilizá-los para realizar uma predição sobre os exemplos de treinamento o metaclassificador.
- Uma nova tabela é criada cujos atributos são os resultados dos classificadores parciais e a classe é o valor original da classe, para cada um dos exemplos de treinamento do metaclassificador.
- Então, sobre essa nova tabela, realizamos o treinamento do metaclassificador.
- Para a etapa de classificação, dado um exemplo, executamos a classificação de cada classificador parcial e, em seguida, executamos o metaclassificador para fazer a predição final sobre o exemplo.

Complexidade x Erro

