

Aprendizado de Máquina

Aula 4.1 - Comitês


Adriano Rivolli

rivolli@utfpr.edu.br

Especialização em Inteligência Artificial


Universidade Tecnológica Federal do Paraná (UTFPR)
Câmpus Cornélio Procopio
Departamento de Computação

Conteúdo

- 
- 1** Introdução
 - 2** Votação
 - 3** Bagging
 - 4** Boosting
 - 5** Stacking

Introdução

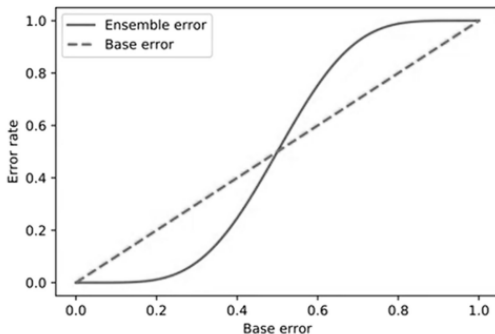
Introdução

- 
- Comitês (*ensemble*) é uma técnica de AM no qual múltiplos modelos preditivos são combinados para melhorar a performance em relação ao uso de um único modelo
 - Permite explorar diferentes vieses de algoritmos
 - Resulta em um modelo mais robusto
 - Os modelos que pertencem ao comitê são chamados de **modelo base**

Diversidade

- Para melhorar a performance é necessário:
 - ▶ Os modelos devem ter taxas de erros similares
 - ▶ A taxa de erro de cada modelo deve ser menor do que 0.5
 - ▶ Os erros de cada modelo devem ser independentes
- Em resumo, é necessário combinar modelos que sejam distintos e que possuam boa taxa de performance

Taxa de erro e a performance do comitê



Fonte: Dolawattha, D.M., Premadasa, H.K.S. and Jayaweera, P.M. (2022), "Evaluating sustainability of mobile learning framework for higher education: a machine learning approach", International Journal of Information and Learning Technology, Vol. 39 No. 3, pp. 266-281. <https://doi.org/10.1108/IJILT-08-2021-0121>

Abordagens

■ Questões:

- ▶ Como combinar os modelos base?
- ▶ Como gerar modelos diferentes?

■ Abordagens:

- ▶ Votação
- ▶ *Bagging (Bootstrap Aggregating)*
- ▶ *Boosting*
- ▶ *Stacking*

Tipos de comitês

■ Homogêneo

- ▶ Todos os modelos são relativos a um único algoritmo

■ Heterogêneo

- ▶ Os modelos são obtidos a partir de diferentes algoritmos de aprendizado

Votação

Votação

- Votação simples (*Hard Voting*)
 - ▶ A predição de cada modelo é um voto em uma classe
- Votação ponderada (*Soft Voting*)
 - ▶ Usa a probabilidade/confiança de cada modelo como peso de um voto em uma classe

Votação simples (exemplo)

■ Problema usando o conjunto de dados Iris:

- ▶ **Membro 1:** virginica
- ▶ **Membro 2:** versicolor
- ▶ **Membro 3:** versicolor
- ▶ **Membro 4:** versicolor
- ▶ **Membro 5:** virginica

■ Resultado:

- ▶ Classe virginica: 2 votos
- ▶ **Classe versicolor:** 3 votos

Votação ponderada (exemplo)

■ Problema usando o conjunto de dados Iris:

- ▶ **Membro 1:** virginica: 0.7, versicolor: 0.3
- ▶ **Membro 2:** virginica: 0.01, versicolor: 0.9
- ▶ **Membro 3:** virginica: 0.1, versicolor: 0.8
- ▶ **Membro 4:** virginica: 0.4, versicolor: 0.6
- ▶ **Membro 5:** virginica: 0.6, versicolor: 0.3

■ Resultado:

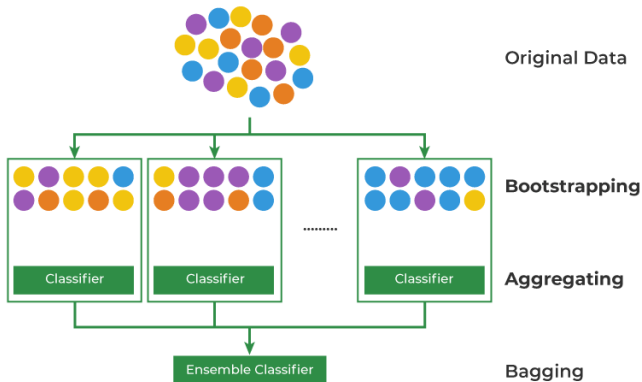
- ▶ Classe virginica: $0.7 + 0.01 + 0.1 + 0.4 + 0.6 = 0.36$
- ▶ **Classe versicolor**: $0.3 + 0.9 + 0.8 + 0.6 + 0.3 = 0.58$

Bagging


Bootstrap Aggregating (Bagging)

- 1 Amostragem Bootstrap
 - ▶ Cria-se múltiplos conjuntos usando amostragem aleatória com repetição (*Bootstrap*)
 - ▶ Uma variação consiste em fazer uma amostragem dos atributos preditivos
- 2 Treinamento dos modelos
 - ▶ Os modelos são independentes entre si (paralelismo)
 - ▶ É possível avaliar os modelos de base e descartar alguns
- 3 Agregação da predição
 - ▶ Usa a votação para gerar a predição final

Bagging (exemplo)



Bagging - Vantagens e desvantagens


- 
- Melhora a performance
 - Robustez a *outliers*
 - Redução da variância
 - Paralelização
 - Flexibilidade
 - Perda de interpretabilidade
 - Computacionalmente caro
 - Não recomendado para algoritmos com baixa variância

Boosting

Boosting

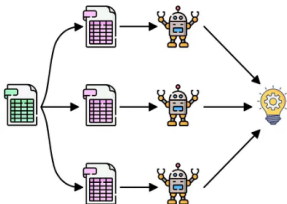
- Uma técnica de AM para melhorar o desempenho de modelos
- Combina modelos "fracos" para formar um modelo unificado "forte"
- Utiliza um processo iterativo que a cada etapa enfatiza as instâncias mais difíceis
- Ao final é possível ponderar os modelos de acordo com a sua respectiva performance

Etapas (AdaBoosting)

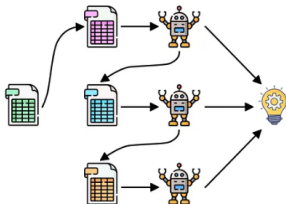
- 
- 1 Atribui o mesmo peso a todas as instâncias
 - 2 Induz um modelo "fraco" que seja melhor que o aleatório
 - 3 Calcular os erros
 - 4 Alterar os pesos aumentar das instâncias erradas e diminuir das corretas
 - 5 Induzir um novo modelo "fraco" usando os novos pesos
 - 6 Repetir as etapas 2-5 até atingir um critério de parada
 - 7 Usa os modelos gerados para fazer a predição

Bagging x Boosting

Bagging



Boosting




Fonte:

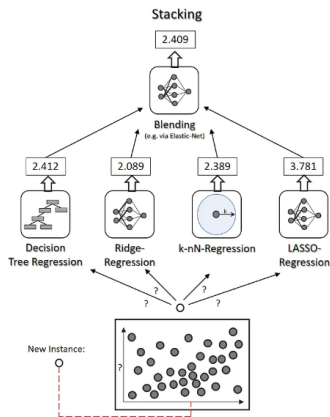
<https://towardsdatascience.com/ensemble-learning-bagging-boosting-3098079e5422>

Stacking


Abordagem de Empilhamento

- 
- Uma abordagem de predição dinâmica que combina a predição de múltiplos modelos
 - A estratégia consiste em treinar um modelo meta (**meta-modelo**) para aprender a combinar as previsões dos modelos base
 - Pode capturar as nuances das previsões dos modelos base

Stacking (exemplo)




Meta-modelo

- 
- Cada modelo base gera uma predição
 - As predições se tornam atributos preditivos da meta-base
 - O meta-modelo aprende a partir destes dados
 - A predição do meta-modelo é a predição final

Meta-base (exemplo)

Exemplo	Árvore	Random Forest	SVM	KNN	Naive Bayes	Classe Real
1	0.7	0.6	0.8	0.5	0.4	1
2	0.4	0.8	0.6	0.7	0.3	0
3	0.9	0.2	0.4	0.6	0.9	1
4	0.2	0.3	0.7	0.4	0.2	0
5	0.6	0.7	0.5	0.3	0.8	1

Predição usando Stacking

- 
- 1 Cada modelo base faz sua predição
 - 2 Gera o meta-exemplo com estes valores
 - 3 O meta-modelo faz a predição do meta-exemplo

Pontos chaves da abordagem

- É uma forma dinâmica de combinar as predições
 - ▶ Não é baseado na votação majoritária
 - ▶ Pode usar a atributos preditivos, predição, probabilidade, combinação destes
- Requer cuidado com vazamento de dados de treinamento (*data leak*)
- A escolha do algoritmo para gerar o meta-modelo é crucial
- Geralmente é utilizado em comitês heterogêneos