

## Combinação de Classificadores - Lista 3 - Poda

Nome: Pedro Diamel Marrero Fernández.

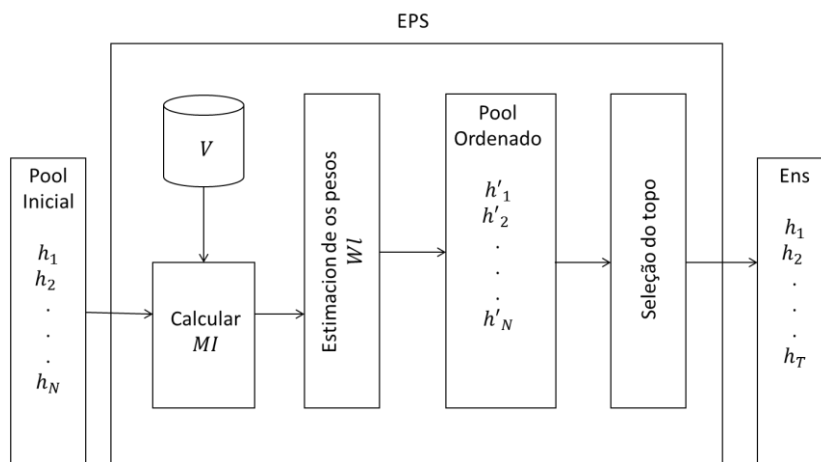
### QUESTÃO 1

Para comparar um pool de classificadores antes e após a poda, podem ser usadas duas alternativas: avaliando o desempenho ou a diversidade [3]. Para avaliar o desempenho podem usar-se, accuracy, ROC, erro. Para avaliar a diversidade podem utilizar-se as medidas estudadas. As alternativas podem ser avaliadas num pool artificial no que se conheça de antemão suas características. Neste caso se poderá analisar se o comportamento da alternativa selecionada é apropriado para um método de poda. Das propostas existentes, as medidas de desempenho são as mais usadas.

### QUESTÃO 2

O algoritmo proposto realiza um análises do espaço de soluções, partindo do subespaço das soluções nas quais cada par de classificadores coincidem e acertam. Posteriormente move-se por esse espaço tentando de criar um ensemble que abrange todas as soluções corretas. Para isto, calcula-se um ranking dinamicamente, que depende do número de objetos que cada par de classificadores é capaz de classificar.

Como observa-se na Fig. 1, a arquitetura proposta parte do cálculo da matriz  $MI$  e realiza a exploração para definir o peso de cada classificador  $Wl$ . Finalmente ordena os classificadores e fica com os  $T$  primeiros. O Algoritmo 1 mostra seus passos principais.



**Figura 1.** Arquitetura do método de poda EPS

---

**Algoritmo 1. EPS - Ensemble Pruning Subspace**

---

Entrada:

*Pool*: pool de classificadores  $\{L_1, L_2, \dots, L_N\}$ .

*V*: conjunto de validação  $\{x_1, x_2, \dots, x_M\}$ .

*C*: classes do conjunto de validação  $\{c(x_1), c(x_2), \dots, c(x_M)\}$ .

*T*: quantidade de classificadores.

Inicializar:

$Ens = \{\}$ ;

$Wl = \{0, 0, \dots, 0\}$ ; vetor de pesos de cada classificador  $L_i$ .

$Wl2 = \{0, 0, \dots, 0\}$ ; vetor de pesos de cada combinação  $L_{i,j}$

Passo 1: Calcular  $R_{M \times N}$  como:  $R_{i,j} = L_j(x_i)$ .

Passo 2: Calcular a matriz da identidade  $MI$ ,

$$MI_{L_{i,j},k} = (L_i(x_k) = L_j(x_k)) \wedge (L_i(x_k) = c(x_k)).$$

Passo 3: Calcular os pesos  $Wl_{L_i}$  de cada classificador  $L_i$ .

Passo 3.1 Se  $Ens \cap Pool = \emptyset$  ir a passo 4.

Passo 3.2 Atualizar os pesos do  $Wl2$  para cada combinação de classificadores, como:

$$Wl2 = Wl2 + \frac{1}{M} \sum_k MI_{L(ij),k}$$

Passo 3.3 Selecionar  $L_{i',j'}$  como:

$$(i', j') = \arg_{i,j} \max(Wl2)$$

Passo 3.4 Fazer:  $Ens = \{L_{i'}, L_{j'}\} \cup Ens$ .

Passo 3.5 Atualizar  $Wl$  como:

$$Wl(Ens) = Wl(Ens) + \max(Wl2)$$

Passo 3.6 Atualizar  $MD$ .

$$MI_{L_{i,j}} = 0,$$

$$MI_{O_{ij}} = MI_{O_{ij}} + \max(MI) + 0,5,$$

onde  $O_{ij}$ , são os objetos para os quais  $MI_{L_{i,j}} = 1$ .

Passo 3.7 Se  $O_{ij}$  tem sido atualizado no passo 3.6 mais de  $\frac{L}{C} + 1$  vezes, fazer:

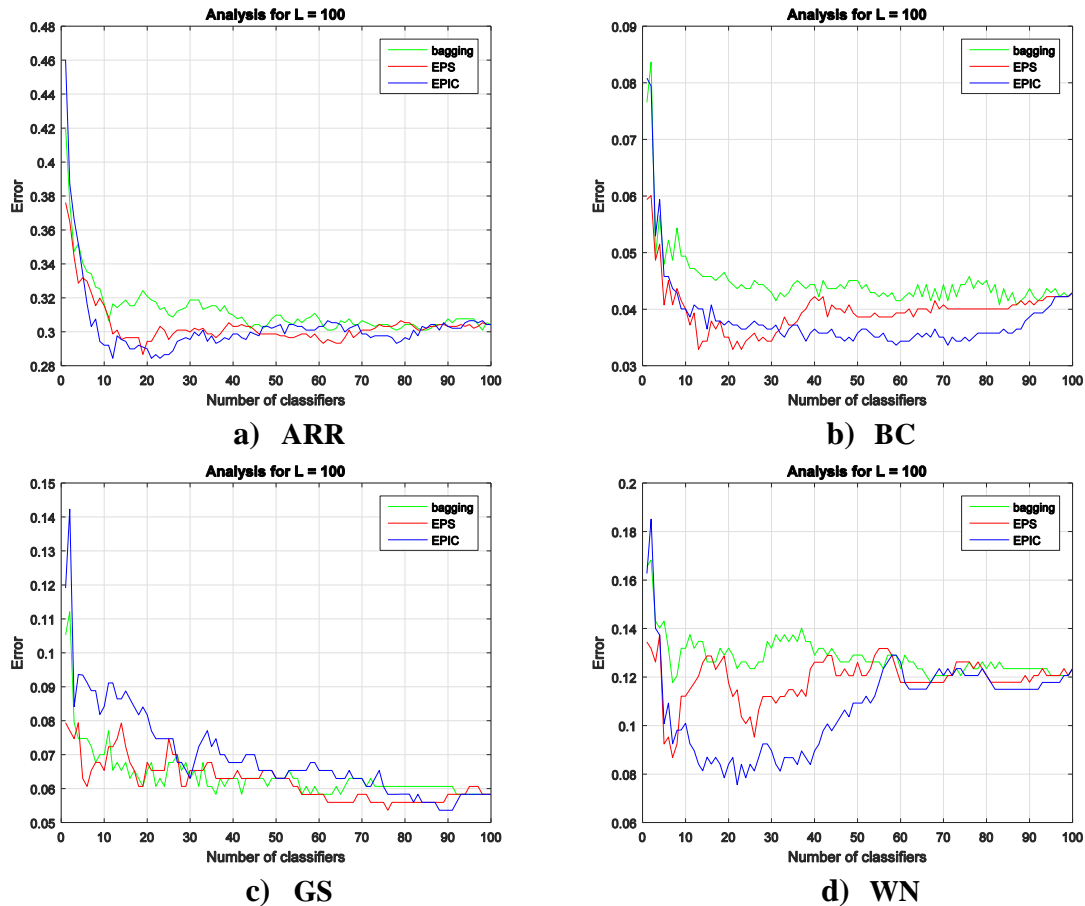
$$MI_{O_{ij}} = 0;$$

Passo 3.7 Ir a passo 3.1.

Passo 4: Ordenar os classificadores pelo peso e retornar os  $T$  primeiros.

---

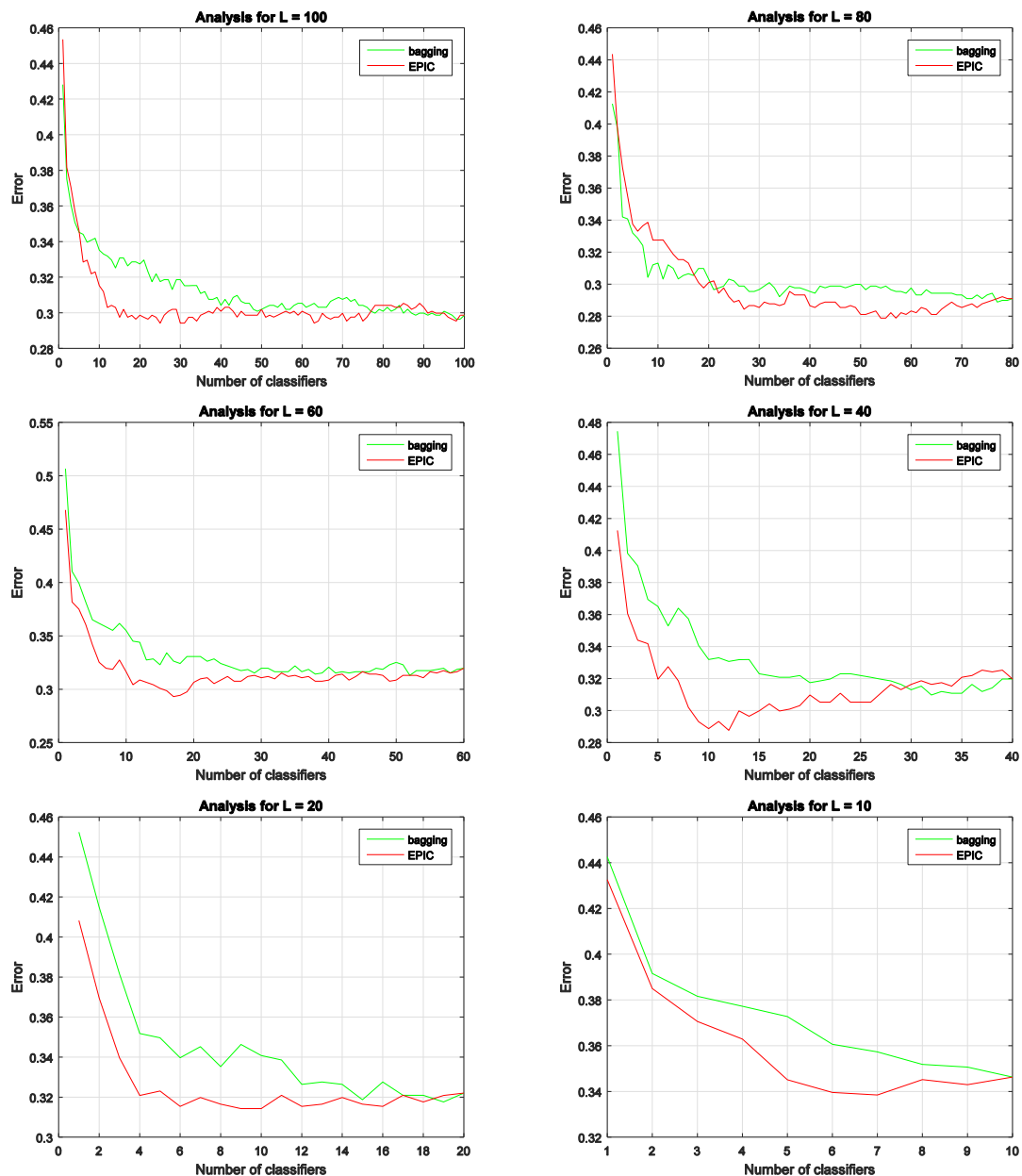
Para a análise dos resultados do método EPS, usou-se 4 bases de dados: *Arrhythmia (ARR)*, *Breast Cancer Wisconsin (BC)*, *Glass (GS)*, *Wine (WN)*, da *UCI Machine Learning Repository* [1]. Utilizou-se as mesmas condições experimentares de [2]. Os resultados se mostram na Fig. 2. O melhor resultado é alcançado usando *BC* com uma redução de um 10% e 30%. Da maneira general os resultados são comparáveis com EPIC e Bagging. Em *GS* os resultados são superiores aos de EPIC. O algoritmo é baseado em ordenação.



**Figura 2.** Resultados comparativos para 4 dataset.

### QUESTÃO 3

Para a análise de EPIC usou-se a base de dados *Arrhythmia (ARR)* da *UCI Machine Learning Repository* [1]. Utilizou-se as mesmas condições experimentares de [2]. A Fig. 3 mostra os experimentos realizados para pool de tamanho 100, 80, 60, 40, 20 e 10. Como se pode observar o comportamento de EPIC para 10%, 20%, 30% es superior ao Bagging.



**Figura 3.** Resultados do método EPIC e Bagging.

## Referências

- [1] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
- [2] Lu, Z., Wu, X., Zhu, X., & Bongard, J. (2010, July). Ensemble pruning via individual contribution ordering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 871-880). ACM.

[3] Partalas, I., Tsoumakas, G., & Vlahavas, I. (2010). An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. *Machine Learning*, 81(3), 257-282.