

# Poda de *pool* de classificadores

Derzu Omaia

*do@cin.ufpe.br*

Universidade Federal de Pernambuco

Professor: George Darmiton da Cunha Cavalcanti

15 de setembro de 2015

# Roteiro

- 1 Poda de pool de classificadores
  - Pool de classificadores
  - Poda de pool de classificadores
  - Objetivos
  - Funciona?
- 2 Métodos de Poda
  - Métodos de poda de pool de classificadores
  - Baseados em Ordenação
  - Baseados em Clusterização
  - Baseados em Otimização
  - Outros Métodos
- 3 Conclusões
- 4 Referências

# Pool de classificadores

- Conjunto de classificadores;
- Geralmente criado por um gerador de *pool* de classificadores;
  - *Bagging*, *Boosting*, *Random Subspace*, *Random Forest*, *Rotation Forest*, e outros métodos.

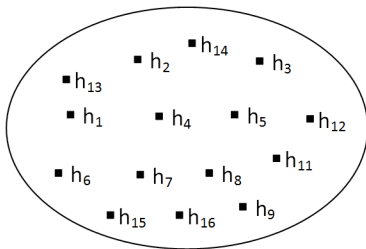


Figura : *Pool* de classificadores.

# Poda

- Reduzir;
- Cortar;
- Diminuir:



Figura : Poda de uma planta.

# Poda de *pool* de classificadores

- Aplicar a Poda sobre um *pool* de classificadores;
- Gera como saída um *ensemble* de classificadores.

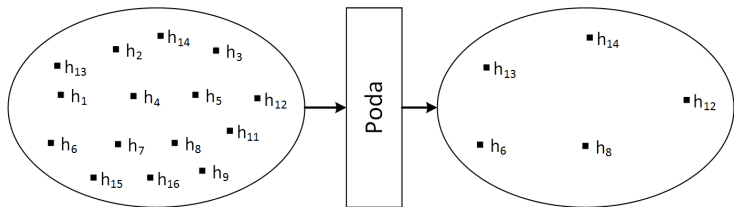


Figura : Exemplo de poda de classificadores.

# Poda de *pool* de classificadores

- Aplicar a Poda sobre um *pool* de classificadores;
- Gera como saída um *ensemble* de classificadores.

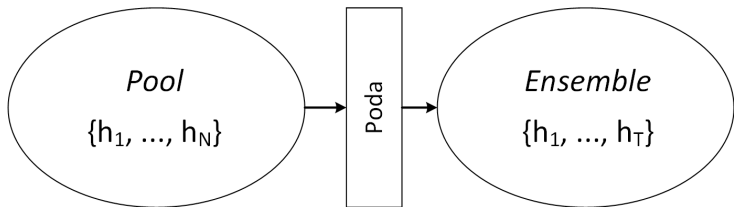


Figura : Exemplo de poda de classificadores.

# Objetivos

- Por que podar?
- Reduzir a quantidade de classificadores envolvidos
  - Reduzir sem sacrificar o desempenho do *ensemble* com todos os membros.
  - Reduz o processamento.
- Excluir maus classificadores;
- Aumentar a diversidade.

# Funciona?

Introduction  
○○○○○○○○○○Classifiers Correlation  
○○●○○○Fusion & Selection  
○○To train or not to train  
○○Final remarks  
○○○○

## MSE Reduction

$$\begin{aligned}
 MSE[F_{comb}] &= E\left[\left(\frac{1}{L} \sum_{i=1}^L \epsilon_i\right)^2\right] \\
 &= \frac{1}{L^2} \sum_{i=1}^L E[\epsilon_i^2] + \frac{1}{L^2} \sum_{i \neq j}^L E[\epsilon_i \epsilon_j] \\
 &= \frac{1}{L} \overline{MSE} + \frac{1}{L^2} \sum_{i \neq j}^L E[\epsilon_i] E[\epsilon_j] \\
 &= \frac{1}{L} \overline{MSE}
 \end{aligned}$$

$$MSE[F_{comb}] = \frac{1}{L} \overline{MSE}$$



# Quando podar?

# Quando podar?

- Quando podar?
  - Quando o desempenho do *ensemble* é superior ao desempenho do *pool*.
  - Quando há correlação entre os classificadores.
  - Medida de desempenho.

$$erro_{poda} \leq erro_{pool}$$

- Teorema MCBTA
  - *Many could be better than all* (Zhou, Wu, & Tang, 2002)

# Roteiro

- 1 Poda de pool de classificadores
  - Pool de classificadores
  - Poda de pool de classificadores
  - Objetivos
  - Funciona?
- 2 Métodos de Poda
  - Métodos de poda de pool de classificadores
  - Baseados em Ordenação
  - Baseados em Clusterização
  - Baseados em Otimização
  - Outros Métodos
- 3 Conclusões
- 4 Referências

# Métodos de poda de *pool* de classificadores

- Categorização (Zhou, 2012):
  - Baseados em Ordenação;
  - Baseados em Clusterização;
  - Baseados em Otimização;
  - Híbridos
  - Outros métodos.

# Métodos de poda de *pool* de classificadores

- Categorização (Zhou, 2012):
  - Baseados em Ordenação;
    - Ordenam classificadores de acordo com algum critério. E seleciona os  $T$  primeiros.
  - Baseados em Clusterização;
    - Clusteriza em grupos e seleciona os protótipos mais representativos para cada grupo.
  - Baseados em Otimização;
    - Considera o problema de poda como um problema de otimização, maximizando ou minimizando um determinado objetivo.
  - Outros métodos.

# Roteiro

- 1 Poda de pool de classificadores
  - Pool de classificadores
  - Poda de pool de classificadores
  - Objetivos
  - Funciona?
- 2 Métodos de Poda
  - Métodos de poda de pool de classificadores
  - **Baseados em Ordenação**
  - Baseados em Clusterização
  - Baseados em Otimização
  - Outros Métodos
- 3 Conclusões
- 4 Referências

# Baseados em Ordenação

- Ordenam classificadores de acordo com algum critério. E seleciona os  $T$  primeiros.

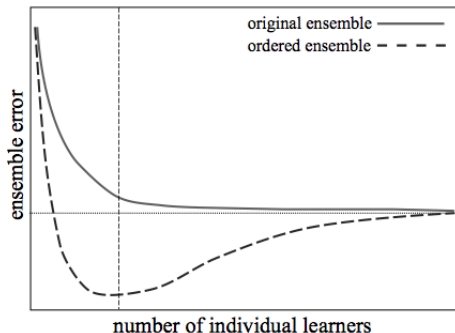


Figura : Curvas de erro do *ensemble* original e do ordenado (Zhou, 2012).

# Baseados em Ordenação

- Métodos:
  - **Kappa Pruning**;
  - Redução de Erro;
  - **Podas Complementares**;
  - Baseado na Orientação;
  - **Baseado em Boosting**;
  - Distância para a borda;
  - **Orientação Ordenada (OO)**;
  - Baseado em Aprendizado por Reforço;
  - **Contribuição Individual Ordenada (EPIC)**



# Kappa Pruning

- Proposto por (Margineantu & Dietterich, 1997);
- Método Clássico;
- Utiliza a medida de diversidade *Kappa-Statistic* ( $\kappa_p$ ) (Cohen, 1960);
- Calcula a diversidade entre todos os pares de classificadores;
- Ordena os pares em ordem decrescente de diversidade;
- Seleciona os  $T$  primeiros do ranking.

# Kappa Pruning

- Arquitetura:

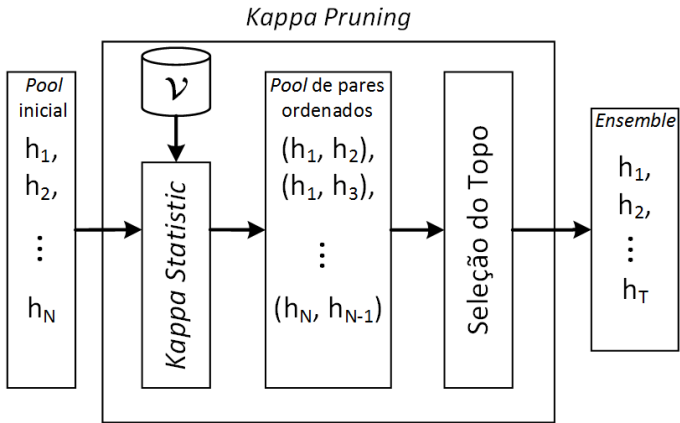


Figura : Arquitetura do método de poda Kappa Pruning

# Kappa Pruning

- Tabela de contingência:

Tabela : Tabela de contingência

	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	b
$h_j = -1$	c	d

- $m = a + b + c + d$

# Kappa Pruning

- Medida de diversidade Kappa-Statistic ( $\kappa_p$ ) (Cohen, 1960);

$$\kappa_p = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2} \quad (1)$$

Sendo  $\Theta_1$  e  $\Theta_2$  as probabilidades de dois classificadores concordarem:

$$\Theta_1 = \frac{a + d}{m} \quad (2)$$

$$\Theta_2 = \frac{(a + b)(a + c) + (c + d)(b + d)}{m^2} \quad (3)$$

# Poda Complementar

- Proposto por (Martínez-Muñoz & Suárez, 2004);
- Inicializa o *ensemble* com o classificador individual que produz o menor erro de validação;
- Iterativamente adiciona mais 1 classificador ao *ensemble*, de forma que o classificador adicionado seja complementar ao *ensemble* atual;
- O classificador,  $h_t$ , adicionado deve maximizar a seguinte condição:

$$h_t = \underset{h_k}{\operatorname{argmax}} \sum_{(x,y) \in \mathcal{V}} \mathbb{I}(h_k(x) = y \text{ and } H_{t-1}(x) \neq y) \quad (4)$$

Sendo:  $H_{t-1}$  o *ensemble* na rodada  $t$ ,  $\mathcal{V}$  o banco de validação,  $\mathbb{I}()$  a função indicadora.

- Seleciona o classificador que obtiver resultado diferente ao do *ensemble*.

# Poda Complementar

- Arquitetura:

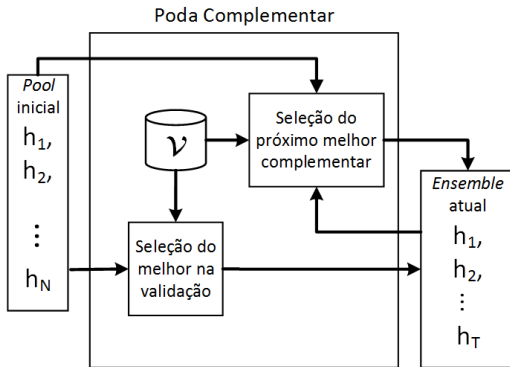


Figura : Arquitetura do método de poda complementar

## Poda baseada em *Boosting*

- Proposto por (Martínez-Muñoz & Suárez, 2007);
- Utiliza AdaBoost para determinar a ordenação dos classificadores individuais.
- Utiliza um vetor de pesos  $w$  do tamanho da quantidade de amostras. Esse vetor é levado em consideração pelo classificador (rede neural).
- Inicializa o *ensemble* com o classificador individual com o menor erro;
- Iterativamente adiciona mais 1 classificador por vez ao *ensemble*, sempre selecionando o de menor erro ponderado;
- Quando o erro for maior que 0.5 o processo de *boosting* é reiniciado (pesos são setados para  $1/N$ , sendo  $N$  a quantidade de amostras);
- Atualiza os pesos aumentando o peso das amostras que produziram erros com o classificador anterior.

# Poda baseada em *Boosting*

- Arquitetura:

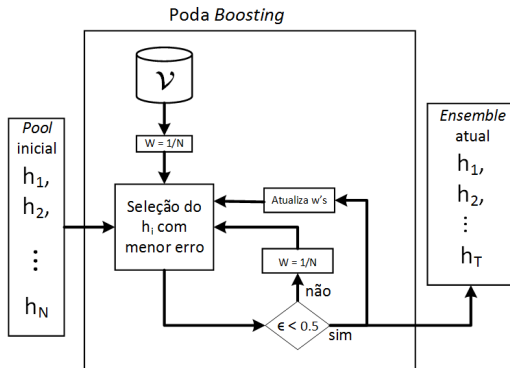


Figura : Arquitetura do método de poda Boosting



# Poda baseada em Orientação Ordenada

## (*Orientation-Ordering*, OO)

- Proposto por (Martínez-Muñoz & Suárez, 2006);
- Cada classificador ( $h_t$ ) é associado a um vetor de assinatura  $c^{(t)}$ , sendo  $t$  o índice do classificador, e  $i$  o índice da amostra;

$$c_i^{(t)} = 2\mathbb{I}(h_t(x_i) = y_i) - 1, \quad (5)$$

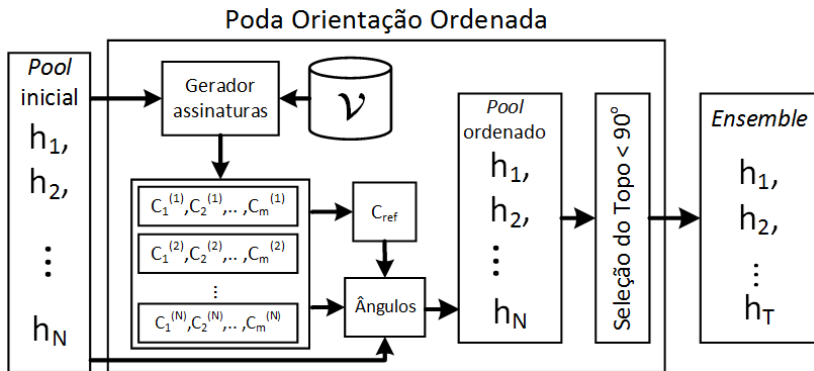
$$\mathbb{I}() \in \{0, +1\}, \text{ então, } c_i^{(t)} \in \{-1, +1\}$$

- Calcula-se o vetor referência ( $c_{ref}$ ) como a média entre todos os vetores;

$$c_{ref} = \frac{1}{N} \sum_{t=1}^N c^{(t)} \quad (6)$$

# Poda baseada em Orientação Ordenada (OO)

- Ordena-se os classificadores em ordem crescente de ângulo em relação a  $C_{ref}$ ;
- Seleciona-se os classificadores com ângulo menor que  $90^\circ$ ;



**Figura :** Arquitetura do método de poda por Orientação Ordenada

# Poda baseada em Contribuição Individual Ordenada (EPIC)

- *Ensemble Pruning via Individual Contribution ordering* (EPIC), proposto por (Lu, Wu, Zhu, & Bongard, 2010);
- Princípios:
  - 1 Quando dois classificadores individuais em um *ensemble* têm precisão similar, o que aumentar a diversidade do *ensemble* deve ter contribuição maior;
  - 2 Quando dois classificadores individuais contribuem de maneira similar para diversidade do *ensemble*, o que tiver a maior precisão deve ter contribuição maior;

# Poda baseada em Contribuição Individual Ordenada (EPIC)

- Calcula a Contribuição Individual (IC) para cada classificador. O IC combina os dois princípios propostos.
- Ordena os classificadores em ordem decrescente de acordo com o IC.
- Seleciona-se os  $T$  classificadores do topo.

# Poda EPIC

- Arquitetura:

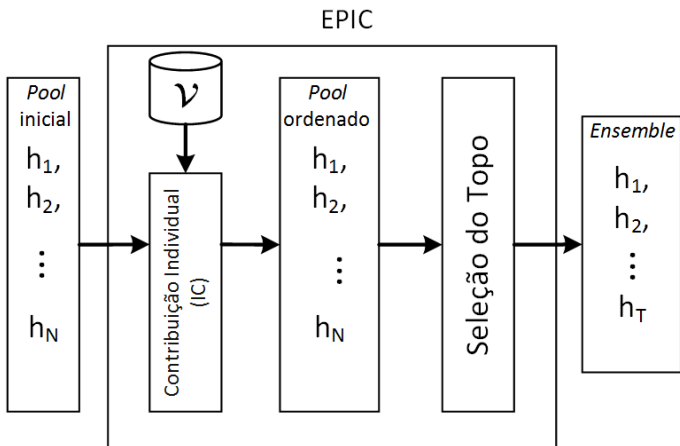


Figura : Arquitetura do método de poda EPIC

# Poda baseada em Contribuição Individual Ordenada (EPIC)

- Contribuição Individual:

$$IC_i = \sum_{j=1}^n IC_i^{(j)} \quad (7)$$

$$IC_i^{(j)} = \begin{cases} 2v_{\max}^{(j)} - v_{h_i(x_j)}^{(j)}, & \text{se } h_i(x_j) = y_j \text{ e } h_i(x_j) \in \text{grp minoritário} \\ v_{\text{sec}}^{(j)}, & \text{se } h_i(x_j) = y_j \text{ e } h_i(x_j) \in \text{grp majoritário} \\ v_{\text{correct}}^{(j)} - v_{h_i(x_j)}^{(j)} - v_{\max}^{(j)}, & \text{se } h_i(x_j) \neq y_j \end{cases} \quad (8)$$

- $j$  amostras,  $i$  classificadores;
- Majoritário: classificador concorda com previsão do *pool*.
- Minoritário: classificador tem resultado diferente da decisão do *pool*.

# Poda baseada em Contribuição Individual Ordenada (EPIC)

- Sendo:

$v_{h_i(x_j)}^{(j)}$  número de classificadores que concordam com  $h_i$ .

$v_{correct}^{(j)}$  número votos corretos para a classe  $y_j$ ;

$v_{max}^{(j)}$  votos recebidos pela classe vencedora para a amostra  $x_j$ ;

$v_{sec}^{(j)}$  votos recebidos pela classe que ficou em segundo lugar para a amostra  $x_j$ ;

# Resultados

- OO vs EPIC;

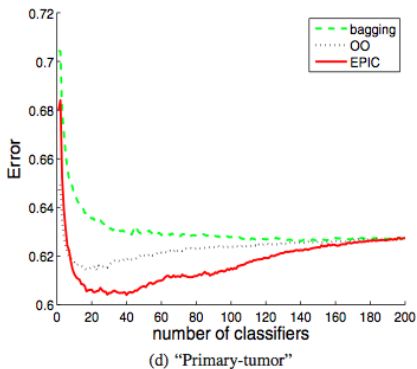
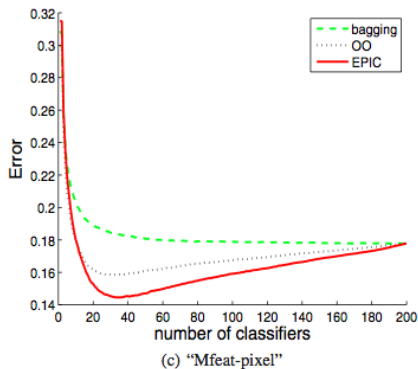


Figura : Resultados obtidos em (Lu et al., 2010)



# Resultados

## • OO vs EPIC;

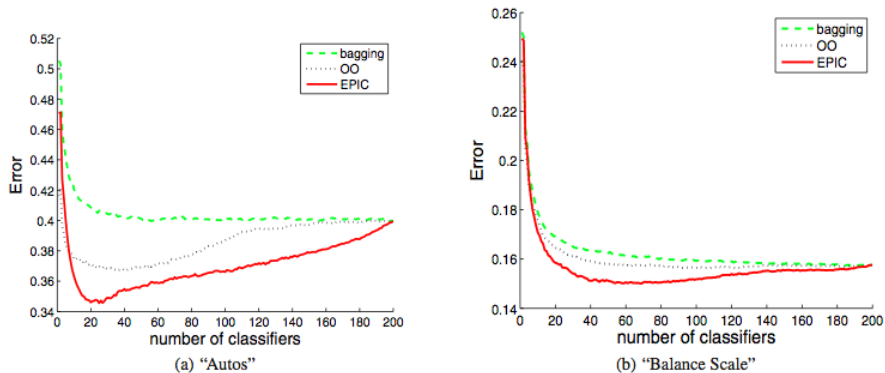


Figura : Resultados obtidos em (Lu et al., 2010)

# Resultados

- OO vs EPIC;

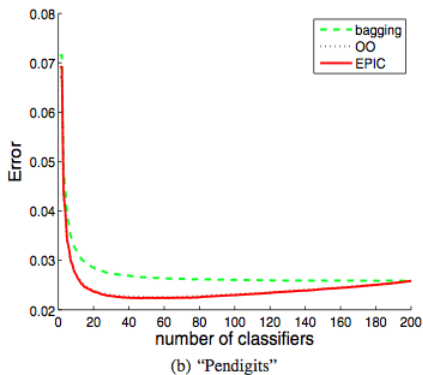
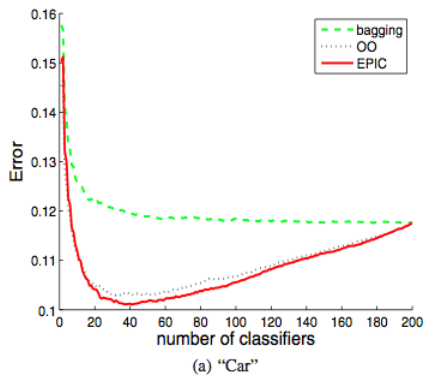


Figura : Resultados obtidos em (Lu et al., 2010)

# Roteiro

- 1 Poda de pool de classificadores
  - Pool de classificadores
  - Poda de pool de classificadores
  - Objetivos
  - Funciona?
- 2 **Métodos de Poda**
  - Métodos de poda de pool de classificadores
  - Baseados em Ordenação
  - **Baseados em Clusterização**
  - Baseados em Otimização
  - Outros Métodos
- 3 Conclusões
- 4 Referências

# Baseados em Clusterização

- Clusteriza em grupos e seleciona os protótipos mais representativos para cada grupo;
- 2 etapas:
  - Clusterização:
    - K-Means;
    - Aglomerado Hierárquico;
    - *Deterministic Annealing*.
  - Seleção de protótipo:
    - Distância entre os classificadores;
    - Remoção iterativa dos menos preciso.

# Roteiro

- 1 Poda de pool de classificadores
  - Pool de classificadores
  - Poda de pool de classificadores
  - Objetivos
  - Funciona?
- 2 **Métodos de Poda**
  - Métodos de poda de pool de classificadores
  - Baseados em Ordenação
  - Baseados em Clusterização
  - **Baseados em Otimização**
  - Outros Métodos
- 3 Conclusões
- 4 Referências

# Baseados em Otimização

- Considera o problema de poda como um problema de otimização, maximizando ou minimizando um determinado objetivo.
- Métodos:
  - Otimização Heurística;
    - **GASEN**;
  - Programação Matemática;
    - **Relaxamento SDP**;
    - Regularização Normal  $\ell_1$ ;
  - Probabilístico:
    - *Expectation Maximization* (EM)
    - *Expectation Propagation* (EP)

# Otimização Heurística

- Método GASEN (*Genetic Algorithm based Selective ENsemble*), proposto por (Zhou et al., 2002):
  - Utiliza algoritmo genético;
  - Os classificadores são associados a pesos que representam a sua aceitabilidade para compor o *ensemble* final;
  - Os pesos de todos os classificadores são armazenados em vetores;
  - Um conjunto de vetores de pesos são inicializados aleatoriamente, gerando a população inicial.
  - A aptidão de cada vetor é calculada baseada na performance do *ensemble* (representado pelo vetor) em um banco de validação;
  - Os classificadores com pesos inferiores a um determinado limiar  $\lambda$  não são utilizados.
  - O resultado final será apenas o vetor que possuir a melhor aptidão.
  - Alto custo computacional.

# Otimização Heurística

- Arquitetura poda GASEN:

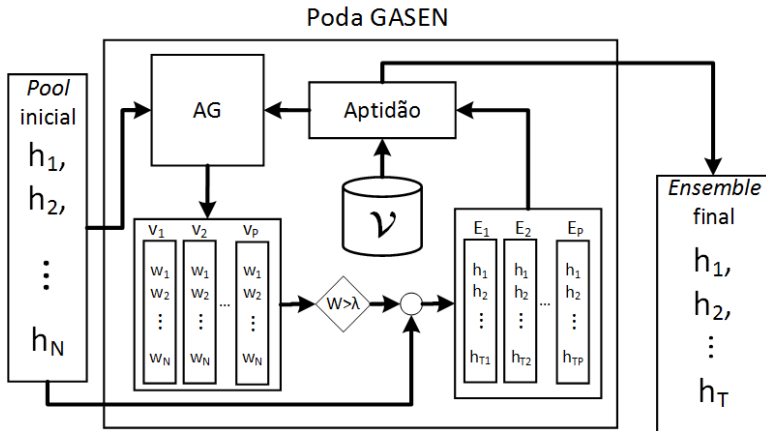


Figura : Arquitetura do método de poda GASEN



# Otimização Heurística

- Método GASEN (*Genetic Algorithm based Selective ENsemble*), proposto por (Zhou et al., 2002):
  - Algoritmo genético tradicional
    - Seleção, mutação, crossover;
  - Condição de parada:
    - Erro não diminuir por 5 gerações;
    - limite de gerações;
    - erro zero;

# Otimização Heurística

- Outros métodos baseados em algoritmos genéticos:
  - Subida da montanha gulosa (*Greedy Hill-climbing*) (Caruana et al., 2004)
  - Algoritmo imunológico artificial (*Artificial immune algorithm*) (Castro et al., 2005, Zhang et al., 2005)
  - Busca de similaridade (*Case similarity search*) (Coyle e Smyth, 2006)

# Otimização por Programação Matemática

- Relaxamento SDP (*Semi-Definitive Programming*):
  - Proposto por (Zhang, Burer, & Street, 2006)
  - Trata o problema da poda como um problema de programação quadrática inteira (*quadratic integer programming*), resolvido em tempo polinomial;
  - É gerada uma matriz  $P_{ij}$ , binária (só 1's e 0's),  $N \times m$ , indicando o acerto (zero) ou o erro (um) de todos os classificadores para todas as amostras ( $N$  classificadores e  $m$  amostras).
  - A matriz quadrada  $G = P * P^T$  tem em sua diagonal o número de erros realizado por cada classificador.

$$P = \begin{bmatrix} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ h_1 & 0 & 1 & 1 & 0 & 0 & 0 \\ h_2 & 0 & 0 & 1 & 0 & 0 & 0 \\ h_3 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad G = P * P^T = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

# Otimização por Programação Matemática

- Relaxamento SDP (*Semi-Definitive Programming*):

- A matriz  $G$  é normalizada gerando  $\tilde{G}$ :

$$\tilde{G}_{ij} = \begin{cases} \frac{G_{ij}}{m} & i = j \\ \frac{1}{2} \left( \frac{G_{ij}}{G_{ii}} + \frac{G_{ji}}{G_{jj}} \right) & i \neq j \end{cases} \quad (9)$$

$$\tilde{G} = \begin{bmatrix} 0.33 & 0.75 & 0 \\ 0.75 & 0.17 & 0 \\ 0 & 0 & 0.50 \end{bmatrix}$$

- Propriedades:

$\sum_{i=1}^N \tilde{G}_{ii}$  mede o desempenho geral do classificador (diagonal);

$\sum_{i,j=1; i \neq j}^N \tilde{G}_{ij}$  mede a diversidade (removendo a diagonal);

$\sum_{i,j=1}^N \tilde{G}_{ij}$  é uma aproximação do erro do *ensemble* (soma tudo).

# Otimização por Programação Matemática

- Relaxamento SDP (*Semi-Definitive Programming*):
  - O problema é formulado como um problema de programação quadrática inteira:

$$\min_z z' \tilde{G} z \quad s.t. \quad \sum_{i=1}^N z_i = T, z_i \in \{0, 1\} \quad (10)$$

Sendo  $z_i$  a indicação se o classificador  $h_i$  faz parte do *ensemble* ou não.  $T$  é o tamanho do *ensemble*.

Desta forma,  $z$  é um vetor binário do tamanho do *pool* inicial que indica quais classificadores farão parte do *ensemble*.

# Otimização por Programação Matemática

- Arquitetura poda Relaxamento SDP:

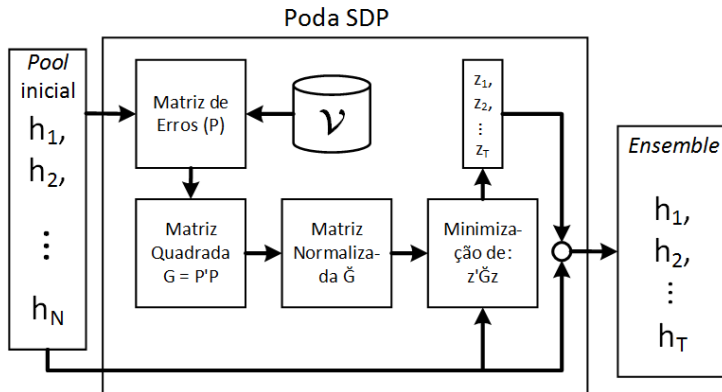


Figura : Arquitetura do método de poda por Relaxamento SDP

# Roteiro

- 1 Poda de pool de classificadores
  - Pool de classificadores
  - Poda de pool de classificadores
  - Objetivos
  - Funciona?
- 2 Métodos de Poda
  - Métodos de poda de pool de classificadores
  - Baseados em Ordenação
  - Baseados em Clusterização
  - Baseados em Otimização
  - **Outros Métodos**
- 3 Conclusões
- 4 Referências

# Outros Métodos

- Métodos desenvolvidos no CIn:
  - Poda utilizando grafos (Carvalho, 2014);
  - Baseado em ordenação (Silva Filho, 2014);



# Conclusões

- Poda vs Seleção;
- A etapa de poda traz vantagens:
  - Diminui custo computacional;
  - Melhora qualidade de *ensemble*;

# Referências I

- Carvalho, G. V. (2014). *Método para poda de pool de classificadores utilizando grafos*. master, Universidade Federal de Pernambuco, UFPE.
- Chen, H., Tino, P., & Yao, X. (2006, Dec). A probabilistic ensemble pruning algorithm. In *Data mining workshops, 2006. icdm workshops 2006. sixth ieee international conference on* (p. 878-882). doi: 10.1109/ICDMW.2006.18
- Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1), 37.

## Referências II

- Li, N., & Zhou, Z.-H. (2009). Selective ensemble under regularization framework. In J. Benediktsson, J. Kittler, & F. Roli (Eds.), *Multiple classifier systems* (Vol. 5519, p. 293-303). Springer Berlin Heidelberg. Retrieved from [http://dx.doi.org/10.1007/978-3-642-02326-2\\_30](http://dx.doi.org/10.1007/978-3-642-02326-2_30) doi: 10.1007/978-3-642-02326-2\_30
- Lu, Z., Wu, X., Zhu, X., & Bongard, J. (2010). Ensemble pruning via individual contribution ordering. In *Proceedings of the 16th acm sigkdd international conference on knowledge discovery and data mining* (pp. 871–880). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/1835804.1835914> doi: 10.1145/1835804.1835914
- Margineantu, D. D., & Dietterich, T. G. (1997). Pruning adaptive boosting. In *lcm1* (p. 211-218).

## Referências III

- Martínez-Muñoz, G., & Suárez, A. (2004). Aggregation ordering in bagging. In *Proc. of the iasted international conference on artificial intelligence and applications* (pp. 258–263). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.146.3650>
- Martínez-Muñoz, G., & Suárez, A. (2006). Pruning in ordered bagging ensembles. In *Proceedings of the 23rd international conference on machine learning* (pp. 609–616). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/1143844.1143921> doi: 10.1145/1143844.1143921
- Martínez-Muñoz, G., & Suárez, A. (2007). Using boosting to prune bagging ensembles. *Pattern Recognition Letters*, 28(1), 156 - 165. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0167865506001802> doi: <http://dx.doi.org/10.1016/j.patrec.2006.06.018>

## Referências IV

- Silva Filho, L. V. (2014). *Uma arquitetura para combinação de classificadores otimizada por métodos de poda com aplicação em credit scoring*. master, Universidade Federal de Pernambuco, UFPE.
- Zhang, Y., Burer, S., & Street, W. N. (2006, December). Ensemble pruning via semi-definite programming. *J. Mach. Learn. Res.*, 7, 1315–1338. Retrieved from <http://dl.acm.org/citation.cfm?id=1248547.1248595>
- Zhou, Z.-H. (2012). Ensemble methods: Foundations and algorithms.
- Zhou, Z.-H., Wu, J., & Tang, W. (2002, May). Ensembling neural networks: Many could be better than all. *Artif. Intell.*, 137(1-2), 239–263. Retrieved from [http://dx.doi.org/10.1016/S0004-3702\(02\)00190-X](http://dx.doi.org/10.1016/S0004-3702(02)00190-X) doi: 10.1016/S0004-3702(02)00190-X