



# **SISTEMAS BASEADOS EM COMITÊS DE CLASSIFICADORES**

**Leandro M. Almeida**

**[leandrolma@gmail.com](mailto:leandrolma@gmail.com)**

# INTRODUÇÃO – PROCESSO DE TOMADA DE DECISÕES

- Procuramos uma segunda, terceira ou quarta opinião.
  - Tratando de assuntos financeiros, médicos, sociais entre outros.
- Atribuímos pesos/valores a cada opinião;
- Com a combinação das opiniões se espera obter uma opinião que seja a mais bem informada de todas;
- O processo de consultar “alguns especialistas” antes de tomar uma decisão é um processo da natureza humana.
- Apenas recentemente o mesmo foi descoberto pela comunidade de inteligência computacional.



# ENSEMBLE BASED SYSTEMS (EBS)

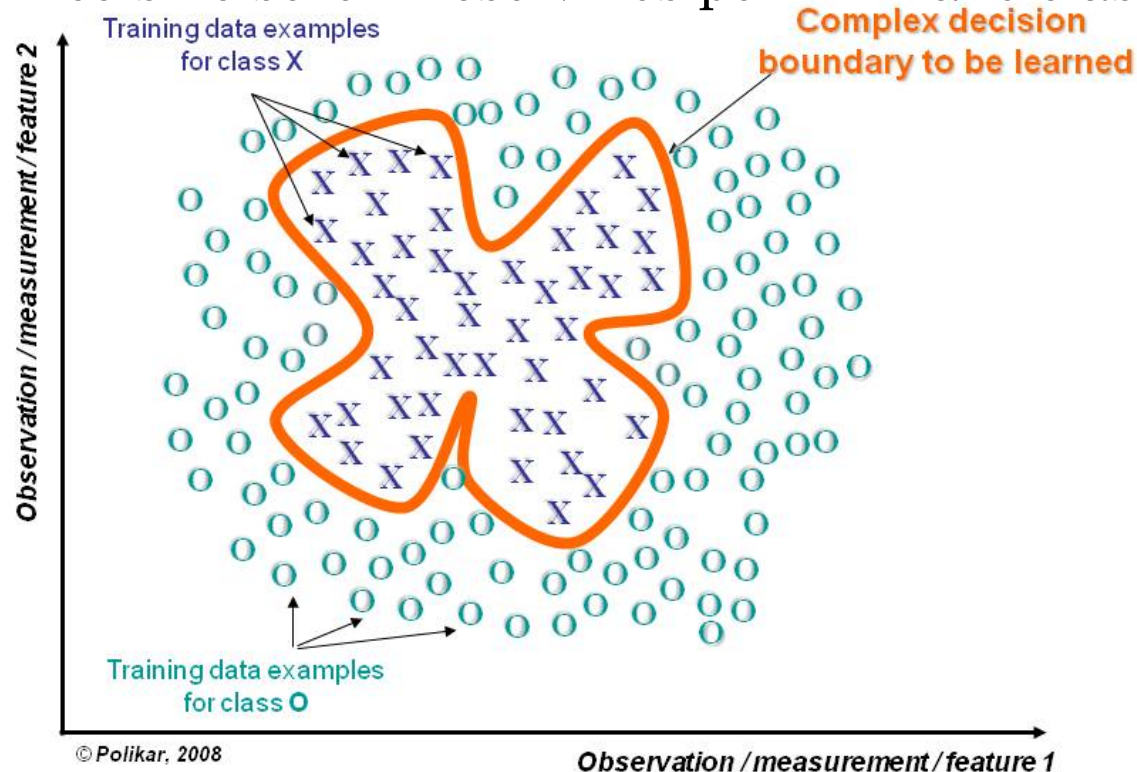
- Também conhecido sobe vários outros nomes:
  - Multiple classifier systems, committee of classifiers, ou mixture of experts.
- Tem mostrado resultados favoráveis comparados a sistemas com um único especialista;
- Bons resultados são encontrados em várias aplicações em uma larga variedade de cenários;
- Projeto, implementação e aplicação de tais sistemas são os principais tópicos desta aula.



# RAZÕES PARA UTILIZAR EBS (CONT.)

## ○ Dividir e Conquistar

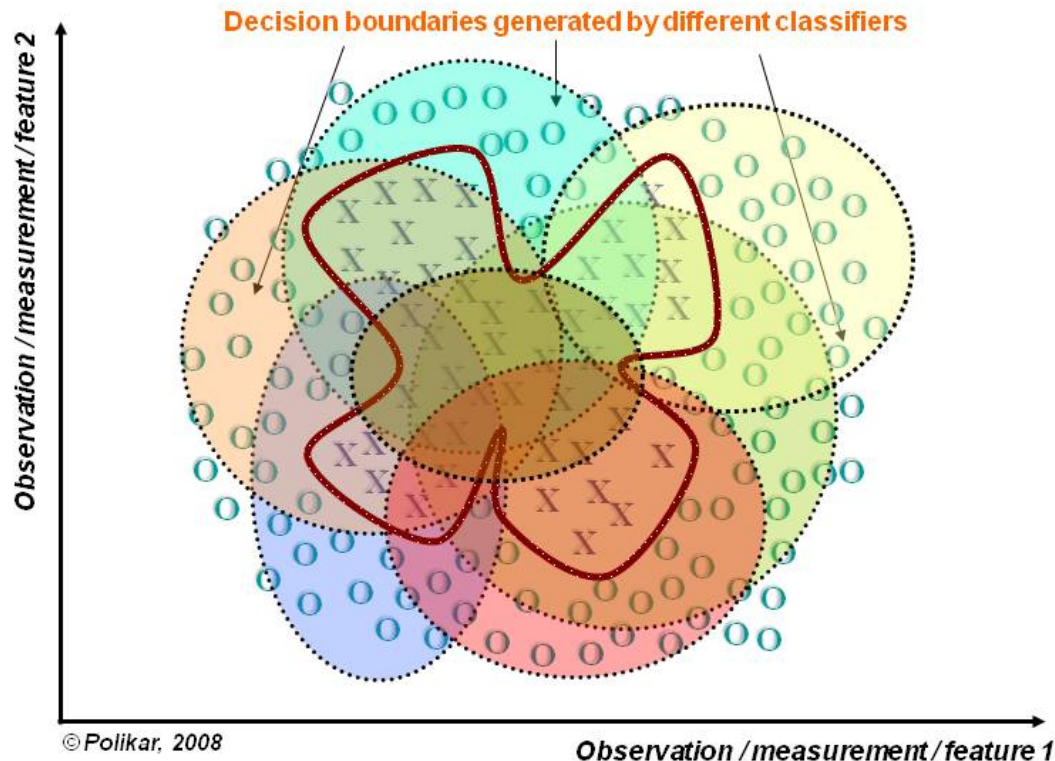
- Independente da quantidade de dados alguns problemas são muito difíceis de serem resolvidos por um dado classificador:

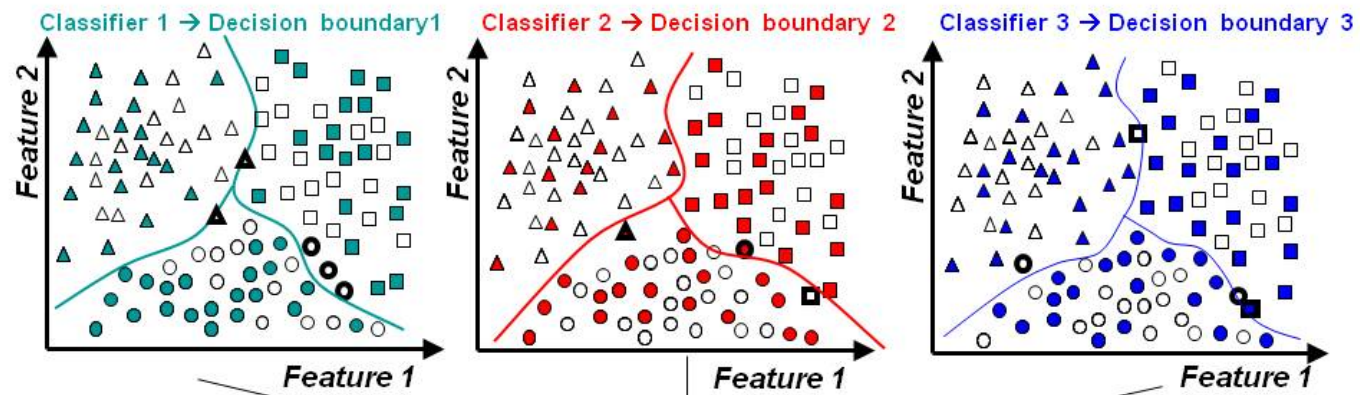


# RAZÕES PARA UTILIZAR EBS (CONT.)

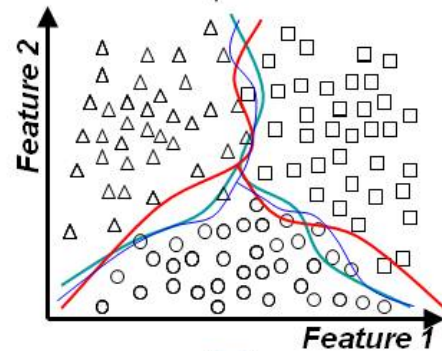
## ○ Dividir e Conquistar

- A fronteira de decisão que separa os dados de diferentes classes pode ser muito complexa ou estar fora do escopo do classificador.

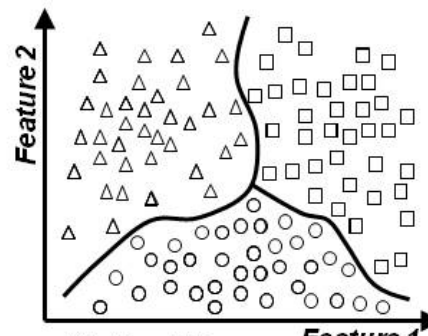




$\Sigma$



Ensemble based decision boundary



# EBS - DIVERSIDADE

- O sucesso de um EBS esta na habilidade de corrigir erros de alguns de seus membros
  - depende fortemente da diversidade dos classificadores que o compõem;
- Cada classificador DEVE fazer diferentes erros em diferentes instâncias dos dados;
- A ideia é construir muitos classificadores e então combinar suas saídas de modo que o desempenho final seja melhor do que o desempenho de um único classificador;
- A diversidade de classificadores pode ser obtida de diferentes formas;



# CRIANDO UM ENSEMBLE

- Como os classificadores serão gerados?
- Como tais classificadores irão diferir entre eles?
- Uma estratégia para geração dos membros de um EBS DEVE buscar uma melhora da diversidade;
- Não existe uma única medida de diversidade aceita uniformemente;
- O aumento da diversidade em EBS é tratado como um problema de busca
  - com emprego de heurísticas - usando procedimentos de resampling ou seleção de diferentes parâmetros de treinamento.





# ALGORITMO BAGGING

- O primeiro algoritmo para a construção de EBS;
- Possui uma implementação simples e intuitiva;
- A diversidade é obtida com o uso de diferentes subconjuntos de dados aleatoriamente criados com reposição;
- Cada subconjunto é usado para treinar um classificador do mesmo tipo;
- As saídas dos classificadores são combinadas por meio do voto majoritário com base em suas decisões;
- Para uma dada instância, a classe que obtiver o maior número de votos será então a resposta.



# ALGORITMO BAGGING

## Algorithm: Bagging

### Input:

- Training data  $S$  with correct labels  $\omega_i$ ,  $\Omega = \{\omega_1, \dots, \omega_C\}$  representing  $C$  classes
- Weak learning algorithm **WeakLearn**,
- Integer  $T$  specifying number of iterations.
- Percent (or fraction)  $F$  to create bootstrapped training data

**Do**  $t=1, \dots, T$

1. Take a bootstrapped replica  $S_t$  by randomly drawing  $F$  percent of  $S$ .
2. Call **WeakLearn** with  $S_t$  and receive the hypothesis (classifier)  $h_t$ .
3. Add  $h_t$  to the ensemble,  $\mathcal{E}$ .

**End**

**Test: Simple Majority Voting** – Given unlabeled instance  $\mathbf{x}$

1. Evaluate the ensemble  $\mathcal{E} = \{h_1, \dots, h_T\}$  on  $\mathbf{x}$ .
2. Let  $v_{t,j} = \begin{cases} 1, & \text{if } h_t \text{ picks class } \omega_j \\ 0, & \text{otherwise} \end{cases}$  be the vote given to class  $\omega_j$  by classifier  $h_t$ .
3. Obtain total vote received by each class,  $V_j = \sum_{t=1}^T v_{t,j}$   $j = 1, \dots, C$ .
4. Choose the class that receives the highest total vote as the final classification.



# ALGORITMO BAGGING - VARIAÇÕES

- Random Forests

- Usado para a construção de EBS com árvores de decisão;
- Variação da quantidade de dados e características;
- Usando árvores de decisão com diferentes inicializações;

- Pasting Small Votes

- Segue a idéia do bagging, mas voltado para grande volumes de dados;
- A base de dados é dividida em subconjuntos chamados de *bites*;
- Trabalha com as instâncias mais informadas.



# ALGORITMO BOOSTING

- Criado em 1990 por Schapire é considerado o mais importante desenvolvimento na história recente da aprendizagem de máquina;
- Também cria EBS por meio da re-amostragem dos dados;
- A re-amostragem é estrategicamente criada para prover o conjunto de treinamento mais informativo para cada classificador;
- Normalmente o EBS possui apenas três classificadores;
- Comprovadamente a performance do EBS é melhor que a performance do melhor indivíduo.



# ALGORITMO BOOSTING

## Algorithm: Boosting

### Input:

- Training data  $S$  of size  $N$  with correct labels  $\omega_i$ ,  $\Omega = \{\omega_1, \omega_2\}$ ;
- Weak learning algorithm **WeakLearn**.

### Training

1. Select  $N_1 < N$  patterns without replacement from  $S$  to create data subset  $S_1$ .
2. Call **WeakLearn** and train with  $S_1$  to create classifier  $C_1$ .
3. Create dataset  $S_2$  as the most informative dataset, given  $C_1$ , such that half of  $S_2$  is correctly classified by  $C_1$ , and the other half is misclassified.:
  - a. Flip a fair coin. If Head, select samples from  $S$ , and present them to  $C_1$  until the first instance is misclassified. Add this instance to  $S_2$ .
  - b. If Tail, select samples from  $S$ , and present them to  $C_1$  until the first one is correctly classified. Add this instance to  $S_2$ .
  - c. Continue flipping coins until no more patterns can be added to  $S_2$ .
4. Train the second classifier  $C_2$  with  $S_2$ .
5. Create  $S_3$  by selecting those instances for which  $C_1$  and  $C_2$  disagree. Train the third classifier  $C_3$  with  $S_3$ .

### Test – Given a test instance $\mathbf{x}$

1. Classify  $\mathbf{x}$  by  $C_1$  and  $C_2$ . If they agree on the class, this class is the final classification.
2. If they disagree, choose the class predicted by  $C_3$  as the final classification.

# ALGORITMO ADABOOST

- O Adaptive Boosting foi criado por Freund and Schapire em 1997;
- É uma versão mais genérica do algoritmo de boosting original;
- Foram criados os AdaBoost.M1 e AdaBoost.R para manipulação de múltiplas classes e para problemas de regressão, respectivamente;
- O AdaBoost gera um conjunto de hipóteses e as combina por meio da votação ponderada;
- As hipóteses são geradas por meio do treinamento de classificadores usando uma distribuição dos dados iterativamente ajustada.



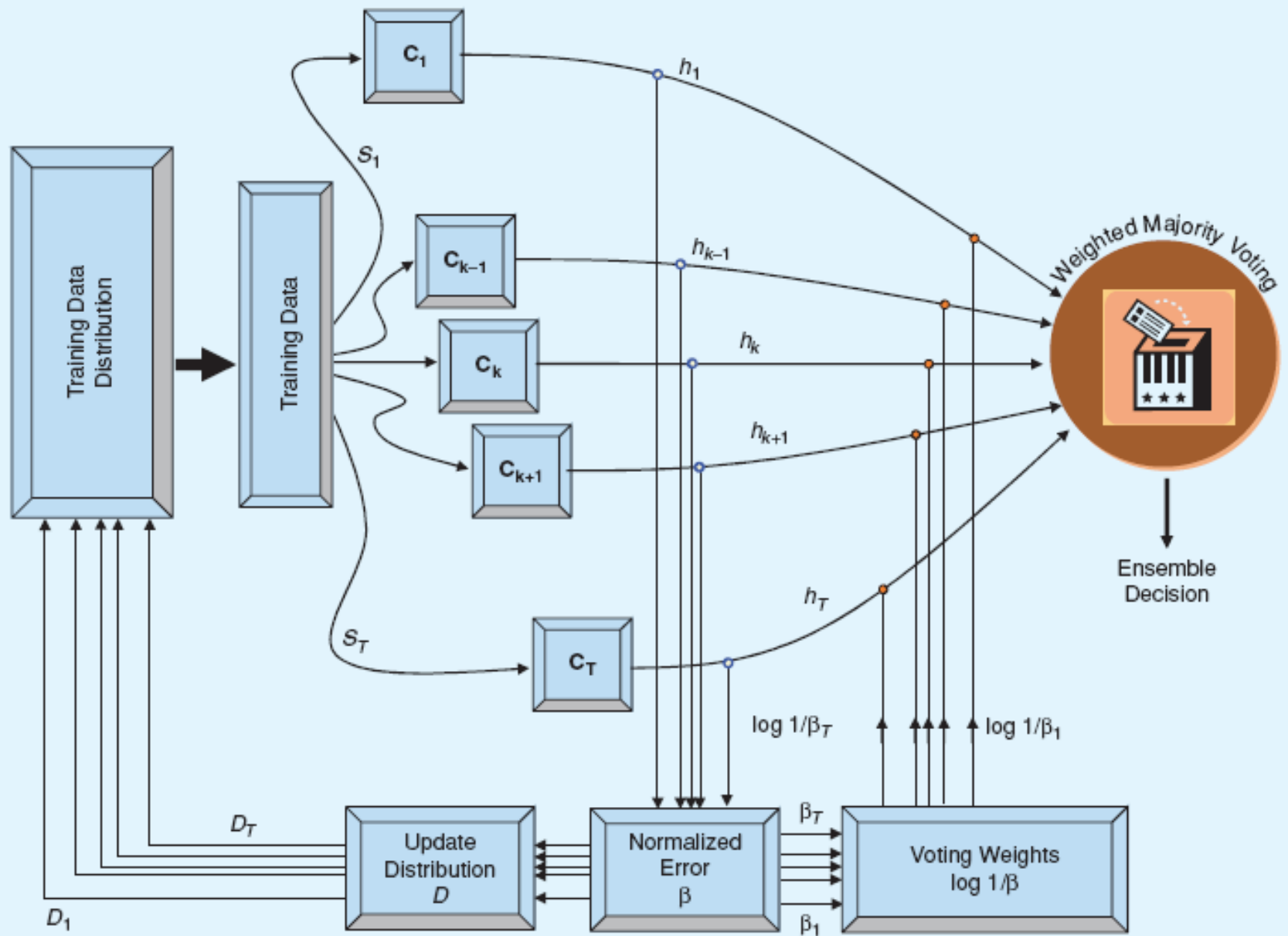


Figure 9. The AdaBoost.M1 algorithm.

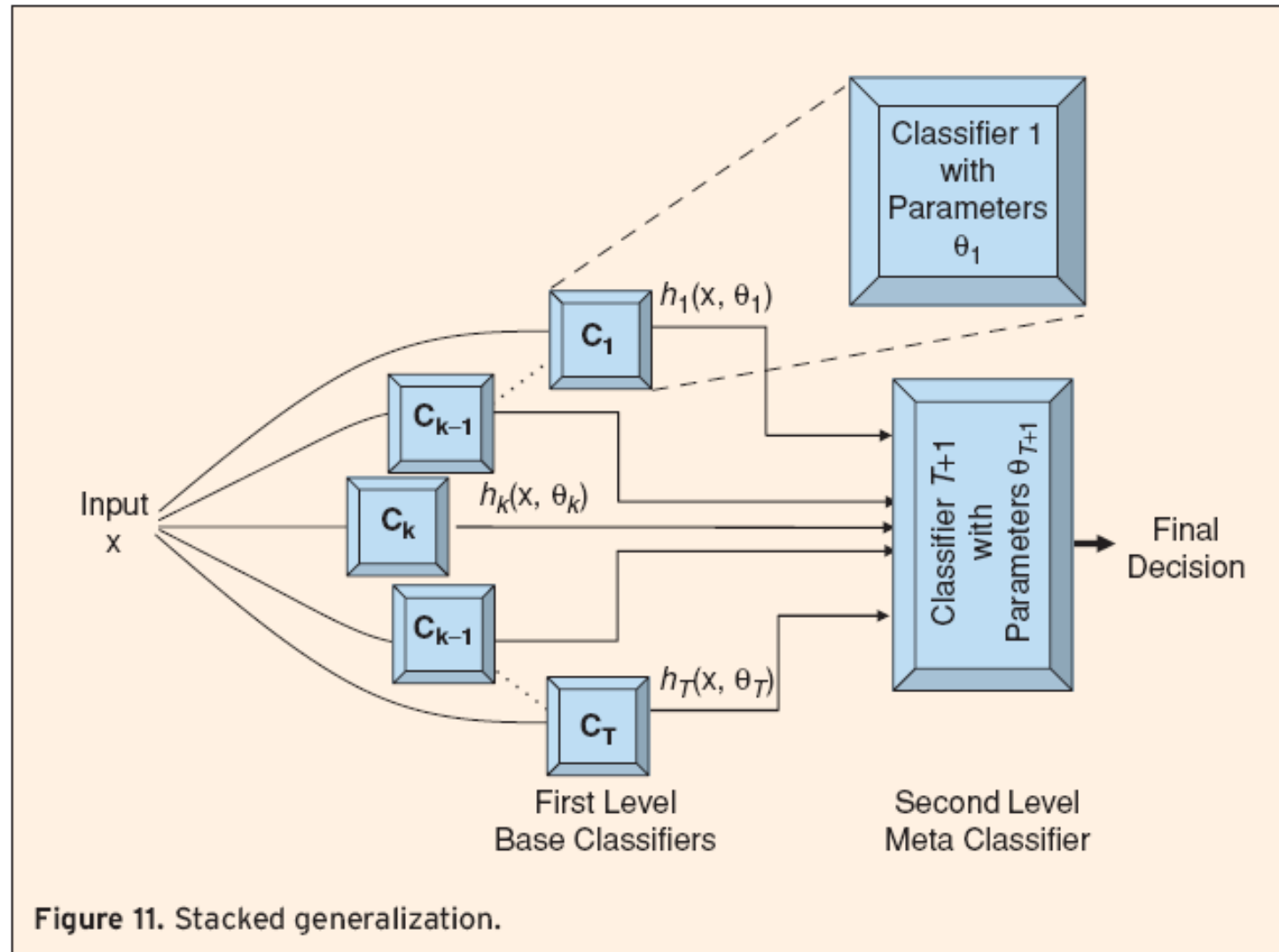
# STACKED GENERALIZATION

- Como aprender a forma de erro e acerto dos classificadores?
- Como mapear as saídas dos classificadores em relação as saídas verdadeiras?
- Os classificadores do EBS são criados usando k-fold, por exemplo;
- As saídas desses classificadores são usadas como entrada para um meta-classificador com o objetivo de aprender o mapeamento entre as saídas e as classes corretas;
- Após o treinamento do meta-classificador os classificadores primários são re-treinados.





# STACKED GENERALIZATION

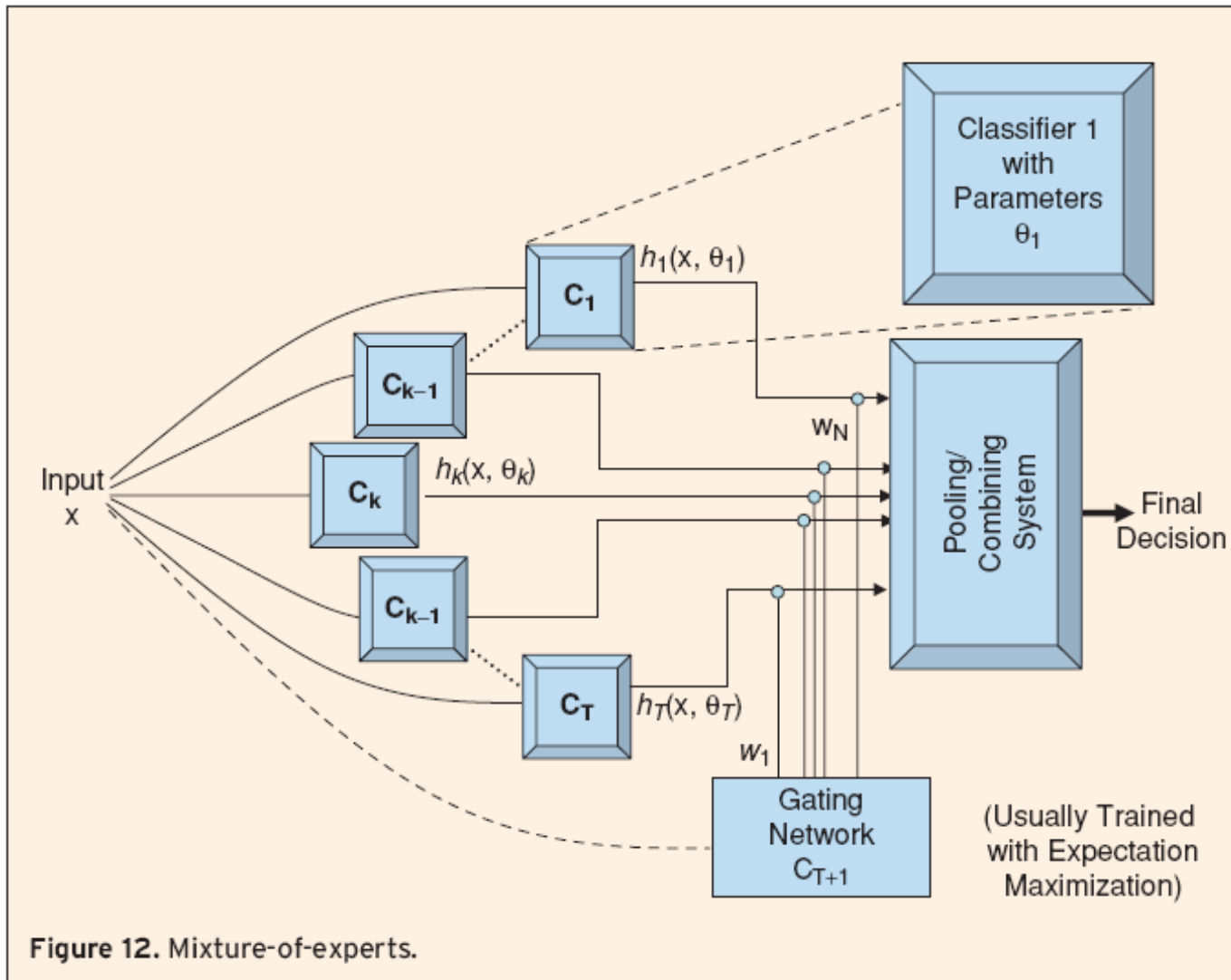


# MIXTURE OF EXPERTS

- Similar ao Stacked Generalization aonde existe um classificador extra ou meta-classificador;
- Neste caso o classificador no segundo nível é usado para atribuir pesos aos classificadores;
- Atualiza a distribuição dos pesos que é utilizada pelo módulo de combinação das decisões;
- O classificador secundário normalmente é uma *gating networks* treinada com gradiente descendente ou Expectation Maximization (EM);
- Tem-se uma regra de combinação dinâmica;
- Os classificadores devem gerar saídas em valores contínuos.



# MIXTURE OF EXPERTS



# MÉTODOS DE COMBINAÇÃO

- Métodos Algébricos
  - Média
  - Média ponderada
  - Soma
  - Soma ponderada
  - Produto
  - Máximo
  - Mínimo
  - Mediana
- Métodos baseados em votação
  - Votação Majoritária
  - Votação Majoritária Ponderada



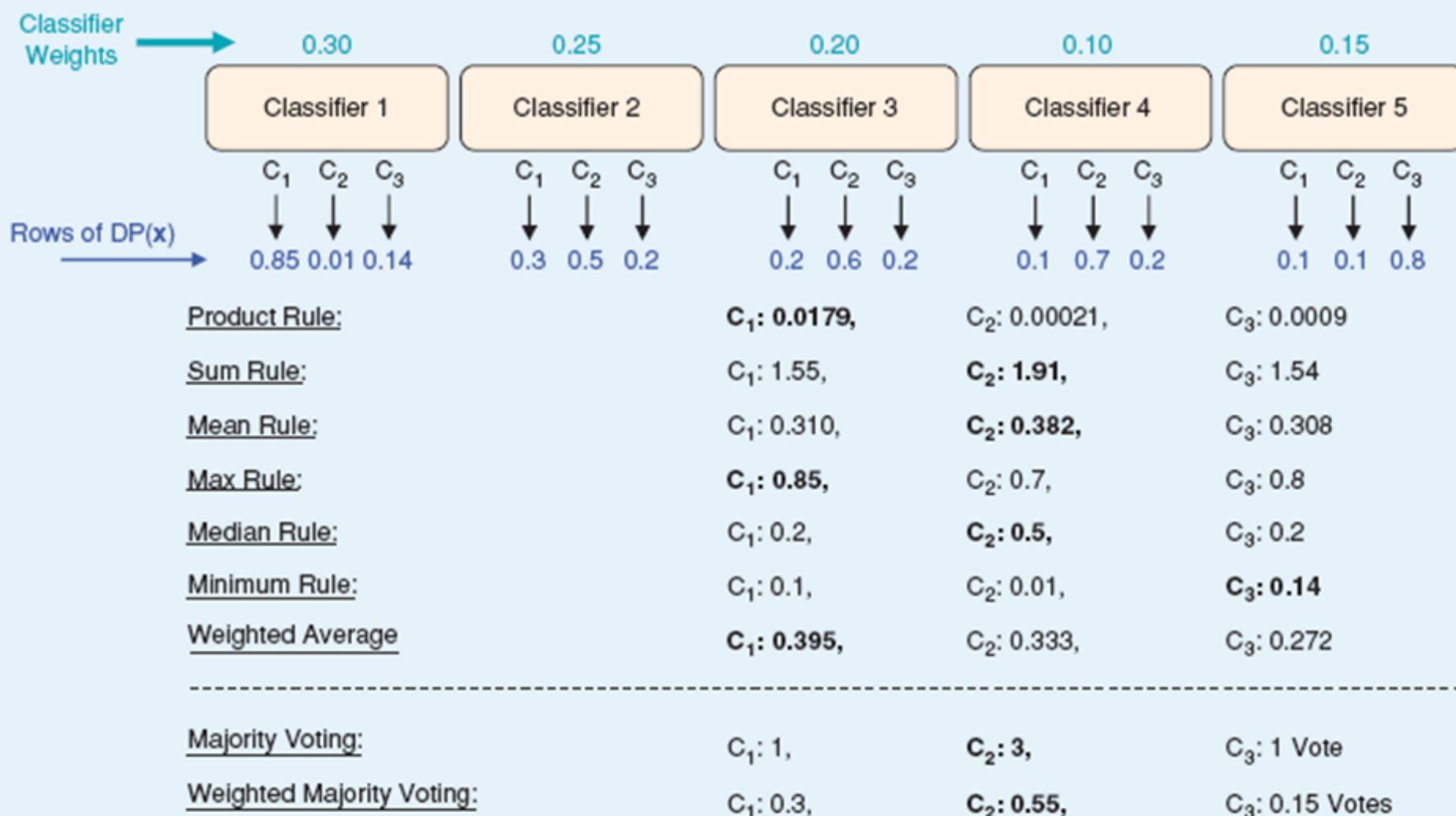


Figure 15. Example on various combination rules.

# EBS - QUESTÕES

- Qual a melhor estratégia para inserção de diversidade?
  - Alguns estudos apontam o método de boosting melhor na média, porém é muito sensível a ruídos e outliers;
- Qual o melhor método de combinação das decisões?
  - No Free Lunch Theorem!**
    - Totalmente dependente do problema a ser resolvido;
    - Muitos preferem o uso da média, devido a sua simplicidade e desempenho consistente;
    - Método baseados em votação são cada vez mais encontrados em trabalhos recentes.



## REFERÊNCIA BIBLIOGRÁFICAS

- S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- R. Polikar, “Ensemble based systems in decision making,” *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, Quarter 2006.
- L. Kuncheva, *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, 2004.
- T. Dietterich, “Ensemble methods in machine learning,” in *Proceedings of the First International Workshop on Multiple Classifier Systems*. London, UK: Springer-Verlag, 2000, pp. 1–15.



# ENSEMBLES - SCRIPTS EM PYTHOM

- Existem os seguintes scripts:
  - Voting\_Ensemble\_Classification.py
    - Comitê com votação entre os classificadores
  - AdaBoost\_Classification.py
    - Utilizando o método AdaBoost
  - random-forest.py
    - Comitê aleatório somente com árvores de decisão
  - bagging-decision-tree.py
    - Comitê de árvores de decisão utilizando o método bagging

