Pruning an ensemble of classifiers via reinforcement learning

Authors: Ioannis Partalas, Grigorios Tsoumakas, Ioannis Vlahavas

Journal: Neurocomputing 72 (2009) 1900-1909

Presentation: Jose Manuel Lopez Guede

Introduction I

- **Ensemble:** a group of predictive models.
- **Ensemble methods:** production and combination of multiple predictive models.
- Used to increase the accuracy of single models.
- They are a solution to:
  - Scale inductive algorithms to large databases.
  - Learn from multiple physically distributed datasets.
  - Learn from concept-drifting data streams (statistical properties of the objective variable change over the time).

Introduction II

- Ensemble methods phases:
  - (1): Production of the different models
    - Homogeneous: from different executions of the same algorithm (changing parameters) on the same dataset.
    - Heterogeneous: from different algorithm s on the same dataset.
  - (2): Combination of the different models
    - Voting, Weighted voting, etc.
  - Recently (1’5): Ensemble pruning: reduction of the ensemble size prior to the combination for 2 reasons:
    - Efficiency
    - Predictive performance

Introduction III

- Pruning an ensemble is NP-Complete:
  - Exhaustive search: not tractable with a large number of models.
  - Greedy approaches: fast, but may lead to suboptimal solutions.
- This paper:
  - Uses Q-L to approximate an optimal policy of choosing whether to include or exclude each model from the ensemble.
  - Extensive experiments.
  - Statistical tests.

Background I

- **Reinforcement Learning:**
  - A problem is specified by a MDP: <S, A, T, R>
    - S: states $s_t \in S$
    - A: actions $a_t \in A(s_t)$
    - T: S x A -> S, transition function, new state s_{t+1}
    - R: S -> Real, reward function, $r_{t+1} \in \Re$
    - Maximize the expected return R_t
  - Model of optimal behaviour: infinite-horizon discounted model
    - γ, $0 \leqslant \gamma < 1$: discount factor

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}.$$

Background II

– Episodes: subsequences of actions
  - Terminal state: modeled as absorbing state
  - Absorbing state: only an action that leads back to itself.

– π : S x A->Real. Policy, $\pi(s, a)$ is the probability of taking the action a in the state s.

– $V^{\pi}(s)$: State-value function. Expected discounted return if the the agent starts from s and follows the policy π .

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s \right\}$$

Background III

– $Q^{\pi}(s,a)$: Action-value function. Expected discounted return if the agent starts executing a in state s following the policy π.

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = a, a_t = a\}$$
$$= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = a, a_t = a\right\}.$$

– π^*: optimal policy, maximizes the state-value $V^{\pi}(s)$ for all states, or the action–value $Q^{\pi}(s,a)$ for all state-action pairs.

Background IV

- To learn the optimal policy:
  - V^*: optimal state-value function
  - Q^*: optimal action-value function: expected return of taking action a in state s following the policy π:

$$Q^*(s,a) = E\left\{ r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \middle| s_t = s, a_t = a \right\}$$

- The optimal policy can be defined:

$$\pi^* = arg \max_{a} Q^*(s,a)$$

- Q-L approximated the Q function:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right)$$

Background V

- **Ensemble methods:**
  - (1) Producting the models:
    - Homogenous models:
      - Different executions of the same learning algorithm.
      - Different parameters of the learning algorithm.
      - Injecting randomness into the learning algorithm.
      - Methods: Bagging, Boosting.
    - Heterogeneous models:
      - Different learning algorithms on the same dataset.
      - Example: ANN, k-NN

Background VI

- (2) Combining the models:
  - There is no single classifier that performs significantly better in every classification problem.
  - Some domains need high performance: medical, financial, …
  - Combine different models to overcome individual limitations

Background VII

- "**Voting**": each model outputs a value, and the value with more votes is the one proposed by the ensemble.
- "**Weighted Voting**": it is like "Voting", but each model is weighted.

Let x be an instance and m_i, $i = 1 \ldots k$ a set of models that output a probability distribution $m_i(x, c_j)$ for each class c_j, $j = 1 \ldots n$.

Output of the method $y(x)$ for the instance x:

$$y(x) = arg\max_{c_j} \sum_{i=1}^{k} w_i m_i(x, c_j)$$

where w_i is the weight of the model i.

Background VIII

- "**Stacked generalization**"/"**Stacking**": combines multiple classifiers by learning a meta-level (or level-1) model that learns the correct class based on the decissions of the base-level (or level-0) classifiers.

Related work

- Heuristics to calculate the benefit of adding a classifier to an ensemble.
- Stochastic search in the space if model subsets with a genetic algorithm.
- Pruning using statistical procedures.
- Generation of 1000 models and pruning.
- …

Our approach I

- **Problem**: pruning an ensemble of classifiers

$$C = \{c_1, c_2, \ldots, c_n\}$$

- Ensemble pruning as a RL task:
  - **States**: pair (C', c_i)

    C' : current ensemble, subset of C.

    c_i : classifier under evaluation.

    State space: $S = P(C) \times C$. P(C): powerset.
  - **Actions**: in each state, there are only 2 actions

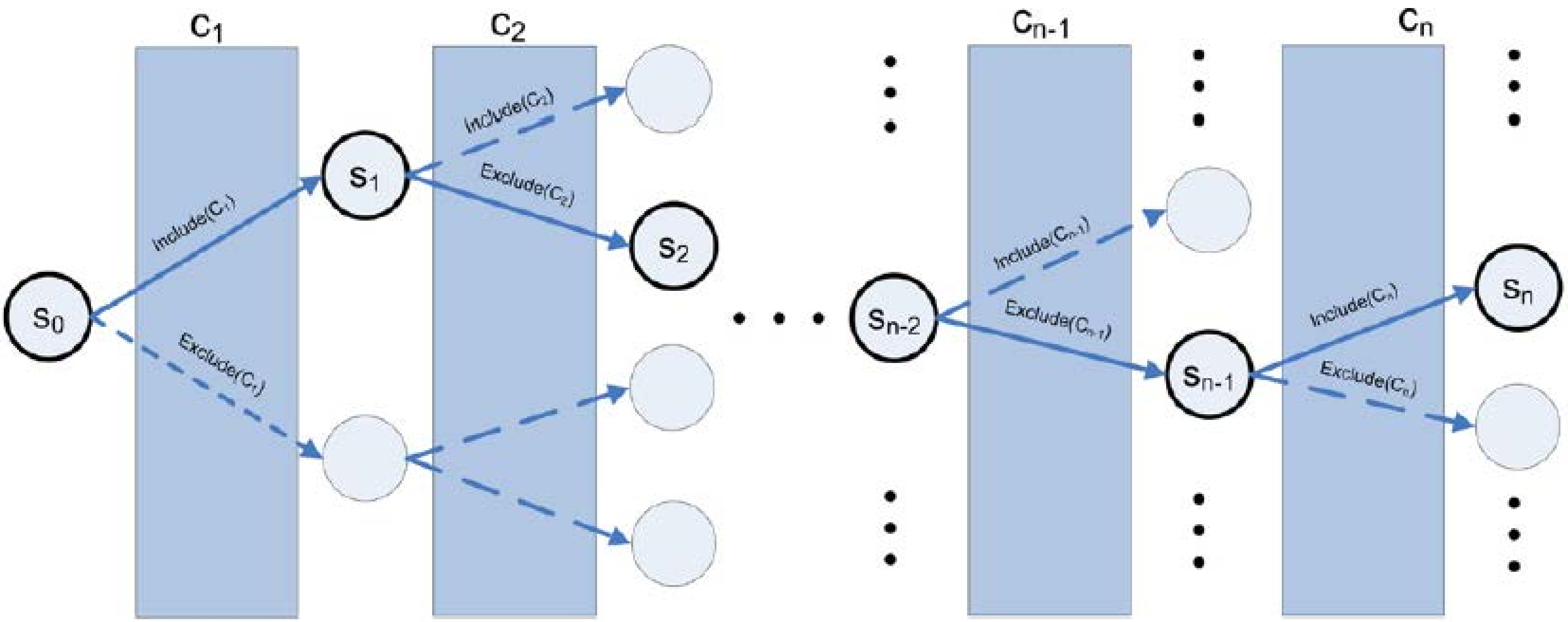
    (Total: 2n actions).

$$A = \bigcup_{i=1}^{n} \{include(c_i), exclude(c_i)\}$$

Our approach II

– **Episodes**:
- The task is modeled as an episodic task
- It starts with an empty set of classifiers $s_0 = (\emptyset, c_1)$
- It lasts n steps.
- At each time step t, the agent chooses to include or not the classifier c_t: $A(s_{t-1}) = \{include(c_t), exclude(c_t)\}$
- End: when the agent arrives at the final state s_n.
- The presentation order of the classifiers is fixed.

Our approach III



Our approach IV

– **Rewards**:
  - Final transition: reward equal to the predictive performance of the ensemble of the final state (intentionally general to be more general).
  - Other transitions: 0
– **Objective**: maximize the performance of the final proned ensemble.

Our approach V

- **The proposed algorithm**:
  - ε –greedy action selection method:

$$a = \begin{cases} \text{a random action with probability } \varepsilon, \\ arg \max_{a'} Q(s, a') \text{ with probability } 1 - \varepsilon. \end{cases}$$

Our approach VI

Pending idea

– Function approximation methods:

- To tackle the problem of large state space.
- Fill the values for every state-action pair in tabular form.
- $Q_t(s, a)$ is a linear function of a parameter vector $\boxed{\vec{\theta}_t}$ (number of parameters equal to the number of features in the state).
  - Training phase: ANN
  - Input: vector with the features of the state. ¿only?
  - Output: estimation of the action value of the state.
  - Feature vector $\vec{\phi}$:
    - » First n coordinates represent the presence or the absence of a classifier.
    - » The last coordinate represent the classifier that is being tested.

¿weights of the ANN?

(1) Initialize $\overrightarrow{\theta}$, $\overrightarrow{e_0} = \overrightarrow{0}$, ϵ and feature vector $\overrightarrow{\phi} = \overrightarrow{0}$

(2) Repeat (for each episode):

   (a) $s \leftarrow (\emptyset, c_1)$

   (b) Estimate $Q(s, include(c_1))$ and $Q(s, exclude(c_1))$

   (c) For each algorithm c_i

      (i) $p \leftarrow RandomReal(0, 1)$

      (ii) If $p < \epsilon$ then

         (A) $a \leftarrow$ random action

         (B) $\overrightarrow{e_0} = \overrightarrow{0}$

      (iii) Else

         (A) $a \leftarrow \arg\max_{a'} Q(s, a')$

         (B) $\overrightarrow{e} = \lambda \overrightarrow{e}$

   (d) $\overrightarrow{e} = \overrightarrow{e} + \phi_s$

   (e) Estimate $Q(s, a), \forall a \in A(s')$

      - $Q(s, a) = \sum_{i=1}^{n+1} \theta(i)\phi_s(i)$

   (f) $a' \leftarrow \arg\max_a Q_a$

   (g) $\delta \leftarrow r + \arg\max_{a'} Q_{a'} - Q_a$

   (h) $\overrightarrow{\theta} \leftarrow \overrightarrow{\theta} + \alpha\delta\overrightarrow{e}$

   (i) Set $s \leftarrow s'$

(3) Until $\overrightarrow{\theta}$ converges.

How is it initilized?

How is it defined?

(1) Initialize $\boxed{\overrightarrow{\theta}}$, $\boxed{\overrightarrow{e_0}} = \overrightarrow{0}$, ϵ and feature vector $\overrightarrow{\phi} = \overrightarrow{0}$

(2) Repeat (for each episode):

(a) $\boxed{s} \leftarrow (\emptyset, c_1)$

What for?

(b) Estimate $Q(s, include(c_1))$ and $Q(s, exclude(c_1))$

Where is it completed?

(c) For each algorithm c_i

(i) $p \leftarrow RandomReal(0, 1)$

(ii) If $p < \epsilon$ then

(A) $a \leftarrow$ random action

(B) $\boxed{\overrightarrow{e_0}} = \overrightarrow{0}$

It is never read

(iii) Else

(A) $a \leftarrow \arg\max_{a'} Q(s, a')$

(B) $\overrightarrow{e} = \boxed{\lambda \overrightarrow{e}}$

How are they defined?

How arethey initialized?

(d) $\overrightarrow{e} = \overrightarrow{e} + \boxed{\phi_s}$

How is defined?

(e) Estimate $Q(s, a), \forall a \in A(s')$

$\boxed{-}$ $Q(s, a) = \sum_{i=1}^{n+1} \theta(i)\phi_s(i)$

¿?

(f) $a' \leftarrow \arg\max_a Q_a$

(g) $\boxed{\delta} \leftarrow r + \arg\max_{a'} \boxed{Q_{a'} - Q_a}$

How is it defined?

It needs the state s to be indexed

(h) $\overrightarrow{\theta} \leftarrow \overrightarrow{\theta} + \boxed{\alpha}\delta \overrightarrow{e}$

Which is its value?

(i) Set $s \leftarrow \boxed{s'}$

It is not written

(3) Until $\overrightarrow{\theta}$ converges.

At the end of each episode, the ensemble is evaluated. Where is it?

Where is the updating rule?

Where is the discount factor?

Experimental setup I

- 20 datasets from the UCI repository.

Table 1
Details of the datasets

UCI folder	Inst.	Cls.	Cnt.	Dsc.	MV (%)
audiology	226	24	0	69	2.03
breast-cancer	286	2	0	9	0.35
breast-cancer-wisconsin	699	2	9	0	0.25
chess (kr-vs-kp)	3196	2	0	36	0.00
cmc	1473	3	2	7	0.00
dermatology	366	6	1	33	0.01
ecoli	336	8	7	0	0.00
glass	214	7	9	0	0.00
heart-disease (hungary)	294	5	6	7	20.46
heart-disease (switzerland)	123	5	6	7	17.07
hepatitis	155	2	6	13	5.67
image	2310	7	19	0	0.00
ionosphere	351	2	34	0	0.00
iris	150	3	4	0	0.00
labor	57	2	8	8	35.75
iymphography	148	4	3	15	0.00
pima-Indians-diabetes	768	2	9	0	0.00
statlog (australian)	690	2	6	9	0.65
statlog (german)	1000	2	7	13	0.00
statlog (heart)	270	2	13	0	0.00

Folder in UCI server, number of instances, classes, continuous and discrete attributes, percentage of missing values.

Experimental setup II

- Each dataset is split into 3 disjuntive parts:
  - D_{Tr} : Training set, 60%.
  - D_{Ev} : Evaluation set, 20%.
  - D_{Te} : Test set, 20%.

Experimental setup III

- Ensemble production methods based on D_{Tr} (weka):
  - 100 homogeneous ensembles:
    - 100 decision trees C4.5 with deafult configuration.
  - 100 heterogeneous ensembles:
    - 2 naive Bayes classifiers
    - 4 decision trees
    - 32 MLPs (multilayer perceptron)
    - 32 k-NN
    - 30 SVMs (support vector machine)
    - Each type of classifiers have been trained with different sets of parameters.

Experimental setup IV

- Once the ensembles have been generated, they are used to compare the EPRL method against:
  - Classifier combination metods:
    - Voting (V)
    - Multiresponse model tresss (SMT)
  - Ensemble pruning methods:
    - Forward selection (FS)
    - Selective fusion (SF)
  - The paper describes the parameters that have been used to train these methods.

Experimental setup V

- **EPRL**:
  - It is executed until the difference in the weights of the ANN between to subsequent episodes becomes less than 10^{-4}.
  - The performance of the pruned ensemble at the end of the episode is evaluated on D_{Ev}, based on its accuracy using voting. ¿?
  - ε: 0.6, reduced by a factor of 0.0001% at each episode
  - λ: 0.9
  - ¿α?

Results and discussion I

- **Heterogeneous case**

To compare multiple algorithms on multiple datasets [Demsar]

Folder in UCI server, accuracy and rank of each method on each of the 20 datasets for the heterogeneous case

UCI folder	Accuracy					Rank				
	FS	EPRL	SF	V	SMT	FS	EPRL	SF	V	SMT
audiology	77.3 ± 4.0	**78.0** ± 4.7	77.8 ± 5.9	75.9 ± 6.1	26.4 ± 5.3	3.0	1.0	2.0	4.0	5.0
breast-cancer	**74.4** ± 4.8	73.3 ± 4.6	71.6 ± 4.2	71.6 ± 4.2	66.5 ± 4.7	1.0	2.0	3.5	3.5	5.0
breast-w	96.3 ± 1.5	96.3 ± 1.6	96.9 ± 1.8	95.0 ± 1.9	**97.5** ± 2.1	3.5	3.5	2.0	5.0	1.0
cmc	52.8 ± 2.4	**53.2** ± 2.7	51.6 ± 4.5	47.1 ± 2.7	45.5 ± 3.6	2.0	1.0	3.0	4.0	5.0
dermatology	96.6 ± 1.5	**96.7** ± 1.5	96.5 ± 1.0	96.4 ± 1.3	65.3 ± 2.2	2.0	1.0	3.0	4.0	5.0
ecoli	**83.9** ± 4.3	82.8 ± 4.8	83.7 ± 5.0	82.4 ± 5.2	67.2 ± 6.1	1.0	3.0	2.0	4.0	5.0
kr-vs-kp	99.3 ± 0.3	99.2 ± 0.2	**99.4** ± 0.2	98.8 ± 0.5	97.6 ± 0.5	2.0	3.0	1.0	4.0	5.0
glass	68.1 ± 5.7	**70.2** ± 6.4	68.6 ± 5.5	68.1 ± 5.5	52.1 ± 7.2	3.5	1.0	2.0	3.5	5.0
heart-h	79.5 ± 5.4	79.0 ± 5.7	79.9 ± 5.6	79.9 ± 5.6	**80.7** ± 6.3	4.0	5.0	2.5	2.5	1.0
hepatitis	81.3 ± 5.9	81.3 ± 5.9	78.1 ± 4.0	78.1 ± 4.0	**81.9** ± 5.9	2.5	2.5	4.5	4.5	1.0
image	96.6 ± 0.6	96.8 ± 0.6	**97.0** ± 0.5	96.2 ± 0.8	64.0 ± 1.0	3.0	2.0	1.0	4.0	5.0
ionosphere	**91.6** ± 3.0	**91.6** ± 3.0	90.7 ± 3.3	83.4 ± 3.2	85.3 ± 3.1	1.5	1.5	3.0	5.0	4.0
iris	94.7 ± 0.4	94.7 ± 0.4	95.7 ± 3.3	94.0 ± 2.4	**99.3** ± 1.3	3.5	3.5	2.0	5.0	1.0
labor	89.1 ± 8.9	89.1 ± 8.9	**94.5** ± 4.5	**94.5** ± 4.5	83.6 ± 7.8	3.5	3.5	1.5	1.5	5.0
lymph	82.4 ± 4.4	80.3 ± 4.3	**85.5** ± 4.8	**85.5** ± 4.8	78.3 ± 6.1	3.0	4.0	1.5	1.5	5.0
diabetes	75.2 ± 4.1	**75.7** ± 3.9	67.5 ± 6.1	66.5 ± 4.6	75.2 ± 4.7	2.5	1.0	4.0	5.0	2.5
credit-a	85.1 ± 1.5	85.5 ± 2.4	**85.7** ± 2.2	83.8 ± 2.3	83.6 ± 3.5	3.0	2.0	1.0	4.0	5.0
credit-g	73.2 ± 2.6	**74.4** ± 2.2	69.0 ± 2.4	69.0 ± 2.4	69.8 ± 2.6	2.0	1.0	4.5	4.5	3.0
heart-statlog	**82.2** ± 5.6	81.9 ± 6.2	81.5 ± 4.3	81.5 ± 3.5	79.1 ± 4.2	1.0	2.0	3.5	3.5	5.0
heart-s	33.3 ± 9.3	32.9 ± 8.6	37.5 ± 8.5	37.5 ± 8.5	**41.3** ± 8.4	4.0	5.0	2.5	2.5	1.0
Average	80.64	80.64	80.43	79.31	72.01	2.575	2.425	2.5	3.775	3.725

Simulated 10 times

Results and discussion II

- EPRL shows its strength and its robustness.
- Next, Friedman's test: compares the average ranks
  - H_0: all algorithms are equivalents.
  - Test F_F based on Friedmans's χ_F^2 statistic
  - With confidence level $p<0.05$, the test allows us to reject the H_0.
- As H_0 has been rejected, Nemenyi test:
  - Post-hoc test intended to find the groups of data that differ after a statistical test of multiple comparisons (such as the Friedman test) has rejected the H_0 that the performance of the comparisons on the groups of data is similar. The test makes pair-wise tests of performance.

Results and discussion III

– As H_0 has been rejected: Nemenyi test:

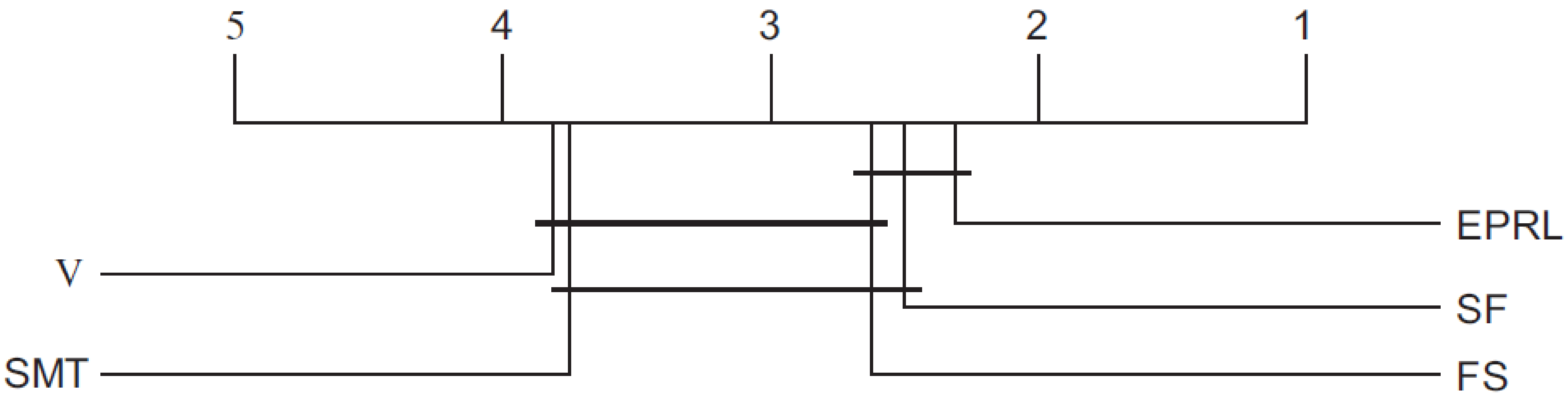


Fig. 3. Graphical representation of the Nemenyi test for the heterogeneous case.

- The algorithms that are not significantly different are connected with a bold line.
- There are 3 groups of similar algorithms.

Results and discussion IV

Table 3
Folder in UCI server and average size of the final ensemble for the heterogeneous case

UCI folder	FS	EPRL	SF
audiology	3.9	3.5	15.6
breast-cancer	3.4	6.7	100.0
breast-w	2.5	3.1	64.8
cmc	11.1	8.6	75.6
dermatology	2.9	1.0	45.5
ecoli	4.1	3.2	57.5
kr-vs-kp	4.2	3.7	42.8
glass	5.0	6.9	79.6
heart-h	2.1	5.2	96.9
hepatitis	1.5	1.9	100.0
image	14.6	9.8	37.0
ionosphere	1.9	3.4	51.0
iris	1.0	1.0	66.6
labor	1.0	1.0	100.0
lymph	2.1	3.8	97.0
diabetes	9.4	10.1	95.7
credit-a	7.1	10.6	71.1
credit-g	9.2	10.4	100.0
heart-statlog	9.3	6.2	74.4
heart-s	3.7	9.3	100.0
Average	5.0	5.47	73.55

Results and discussion V

– Average type of models selected for all datasets:

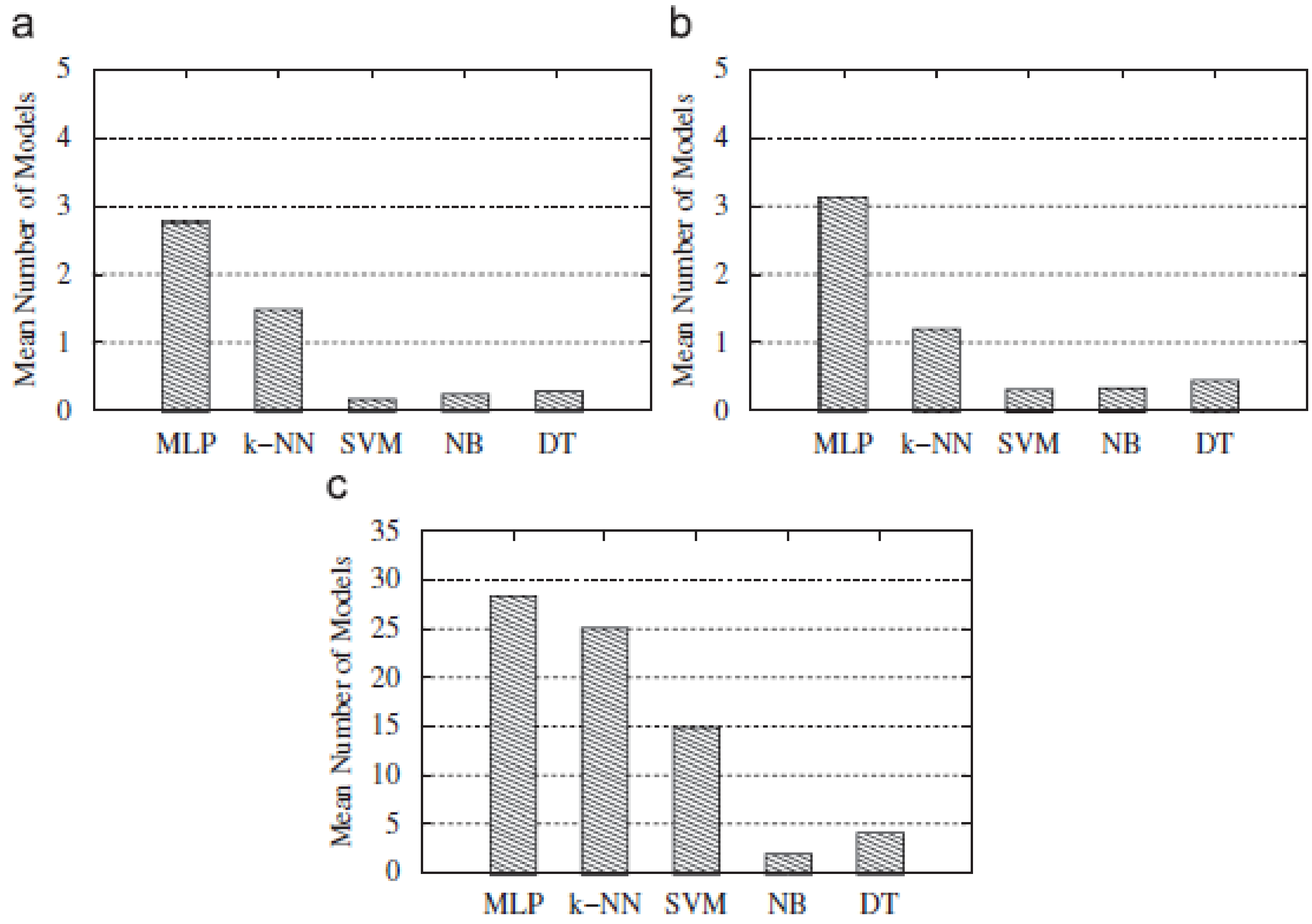


Fig. 4. Type of selected models for each algorithm. (a) FS; (b) EPRL; (c) SF.

Results and discussion VI

- **Homogeneous case**

Table 4
Folder in UCI server, accuracy and rank of each method on each of the 20 datasets for the homogeneous case

UCI folder	Accuracy				Rank			
	FS	EPRL	V	SMT	FS	EPRL	V	SMT
audiology	74.7 ± 6.9	**76.2** ± 3.7	76.1 ± 4.0	60.2 ± 7.8	3.0	1.0	2.0	4.0
breast-cancer	73.9 ± 4.3	73.9 ± 4.8	**76.0** ± 4.1	62.6 ± 4.5	2.5	2.5	1.0	4.0
breast-w	95.5 ± 1.6	95.7 ± 1.5	**95.8** ± 1.1	95.3 ± 1.4	3.0	2.0	1.0	4.0
cmc	52.4 ± 3.1	52.8 ± 2.5	**53.8** ± 2.3	44.3 ± 4.6	3.0	2.0	1.0	4.0
dermatology	94.2 ± 2.1	94.4 ± 2.8	**96.3** ± 3.3	91.9 ± 2.7	3.0	2.0	1.0	4.0
ecoli	83.6 ± 3.7	**85.1** ± 2.4	84.9 ± 2.3	79.7 ± 4.0	3.0	1.0	2.0	4.0
kr-vs-kp	**99.2** ± 0.3	**99.2** ± 0.3	**99.2** ± 0.3	98.8 ± 0.3	2.0	2.0	2.0	4.0
glass	68.6 ± 6.3	67.1 ± 6.0	**71.0** ± 6.8	54.3 ± 6.4	2.0	3.0	1.0	4.0
heart-h	77.6 ± 3.7	**78.4** ± 3.2	77.9 ± 3.1	75.5 ± 4.2	3.0	1.0	2.0	4.0
hepatitis	78.4 ± 5.5	78.4 ± 7.9	**79.4** ± 5.2	77.4 ± 5.7	2.5	2.5	1.0	4.0
image	96.4 ± 0.6	**96.8** ± 0.8	**96.8** ± 0.7	94.1 ± 1.1	3.0	1.5	1.5	4.0
ionosphere	90.6 ± 2.2	90.6 ± 2.6	**93.0** ± 2.4	86.1 ± 3.2	2.5	2.5	1.0	4.0
iris	94.3 ± 3.0	94.0 ± 2.9	**96.3** ± 3.1	94.7 ± 4.2	3.0	4.0	1.0	2.0
labor	72.7 ± 1.1	74.5 ± 1.2	**79.1** ± 1.0	54.5 ± 1.1	3.0	2.0	1.0	4.0
lymph	75.2 ± 7.0	77.6 ± 9.0	**78.3** ± 9.0	65.9 ± 8.2	3.0	2.0	1.0	4.0
diabetes	74.5 ± 3.9	75.0 ± 4.2	**75.3** ± 4.2	67.9 ± 4.4	3.0	2.0	1.0	4.0
credit-a	86.7 ± 2.1	86.9 ± 2.2	**87.3** ± 2.3	83.8 ± 3.1	3.0	2.0	1.0	4.0
credit-g	73.3 ± 2.2	73.6 ± 2.4	**75.2** ± 2.3	67.7 ± 3.1	3.0	2.0	1.0	4.0
heart-statlog	77.2 ± 5.9	80.0 ± 5.4	**81.5** ± 3.7	71.9 ± 4.3	3.0	2.0	1.0	4.0
heart-s	35.8 ± 7.7	41.3 ± 8.0	**42.9** ± 4.9	35.0 ± 8.5	3.0	2.0	1.0	4.0
Average	78.7	79.6	$\boxed{80.8}$	73.1	2.825	2.05	$\boxed{1.225}$	3.9

Results and discussion VII

– Nemenyi test:

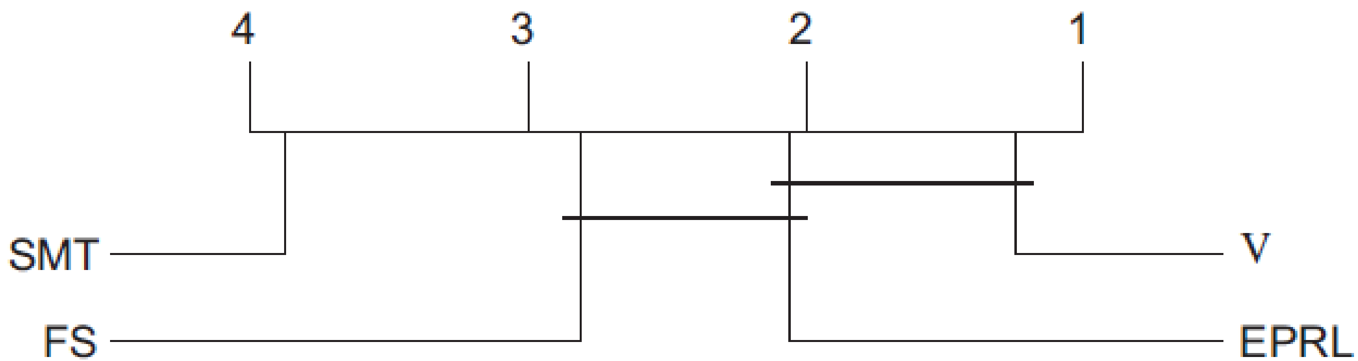


Fig. 5. Graphical representation of the Nemenyi test for the homogeneous case.

- EPRL is in the best group of algorithms.

Results and discussion VIII

Table 5
Folder in UCI server and average size of the final ensemble for the homogeneous case

UCI folder	FS	EPRL
audiology	4.8	5.5
breast-cancer	4.4	7.6
breast-w	4.5	9.1
cmc	13.1	19.7
dermatology	2.4	4.9
ecoli	5.5	10.9
kr-vs-kp	2.5	3.6
glass	5.9	8.2
heart-h	4.9	4.1
hepatitis	2.5	3.7
image	9.7	8.0
ionosphere	3.4	5.3
iris	1.1	4.1
labor	1.5	1.7
lymph	3.5	5.8
diabetes	10.1	11.4
credit-a	5.7	8.7
credit-g	14.7	14.7
heart-statlog	6.7	9.5
heart-s	6.5	12.3
Average	5.67	7.94

Results and discussion IX

- **Running times**
  - Times for the “image” dataset.
  - ¿In which type of machine?

Table 6
Running times of the algorithms for one indicative dataset

FS (min)	EPRL (min)	SF (min)	SMT (min)
0.21	5.35	0.16	0.48

Anytime pruning I

- The proposed approach has the "anytime" property:
  - It can output a solution at any given time point.
  - As the ε parameter becomes small, the exploration ceases and there is only exploitation, without improve.
- It would be desirable that the EPRL continued improving with time: Learning periods.

Anytime pruning II

- Learning period:
  - It consistfs of a number of episodes.
  - When the period starts, ε has a high value, and is decayed over the episodes.
  - It end when ε is less than a small threshold.
- Experimental design:
  - Heterogeneous and Homogeneous models.
  - A learning period begins with ε=0.6, end with ε<0.05 and decays by a factor of 10^{-4}.
  - An interesting idea.

Anytime pruning III

– Four firts periods.

– All datasets:

Table 7
Average rank of all algorithms for the heterogeneous case

Period	FS	EPRL	SF	V	SMT
1	2.775	2.625	2.5	3.8	3.725
2	2.725	2.225	2.55	3.8	3.775
3	2.8	1.95	2.7	3.85	3.775
4	2.85	1.8	2.75	3.85	3.825

Table 8
Average rank of all algorithms for the homogeneous case

Period	FS	EPRL	V	SMT
1	2.7	2.025	1.15	3.9
2	2.8	1.875	1.3	3.95
3	2.8	1.875	1.3	3.95
4	2.8	1.875	1.3	3.95

Conclusions

- A new method for pruning is proposed.
- It get a high predictive performance.
- It produces small sized ensembles.
- It can output a solution anytime.
- Its computational complexity is linear with respect to the ensemble size, but the state space grows exponentially with the number of classifiers.
- Running Time is high.