Versions of Random Forests: Properties and Performances

Choongsooon Bae

Google Inc.
U.C. Berkeley

March 26, 2009

Joint work with Peter Bickel
Outline

Motivation

CART

CART construction
Examples

Bagging

Definition
Comparison

Random Forests

Definition
Breiman’s Random Forests
Purely Random Forest
Bagging averaged 1-nearest neighbor classifier
Data Adaptive Weighted Random Forests

Performances

Example I
Example II
The truth

Goals: Prediction

: Information
Large and High dimensional Data Set

- Internet advertisements data: 3,279 data, 1,558 attributes.
  \[(n = 3,279, d = 1,558)\].

- Microsoft web data: 37,711 data, 294 attributes.
  \[(n = 37,711, d = 294)\].

- Corel Image data: 68,040 images, 89 attributes.
  \[(n = 68,040, d = 89)\].

- Spam E-mail Data: 4,601 data, 57 attribute.
  \[(n = 4,601, d = 57)\].
Issues

- Fast calculation.
- Excellent accuracy.
- Good insights into the inside of black box
Machine Learning Methods

- Kernel smoothing.

- **Classification and Regression Tree** (CART).

- **Support Vector Method** (SVM).

- Boosting.

- Bagging (Bootstrap Aggregating).

- Random Forests.
Outline

**Motivation**

**CART**
- CART construction
- Examples

**Bagging**
- Definition
- Comparison
- Basic Idea and Issues

**Random Forests**
- Definition
- Breiman’s Random Forests
- Purely Random Forest
- Bagging averaged 1-nearest neighbor classifier
- Data Adaptive Weighted Random Forests

**Performances**
- Example I
- Example II
CART

400 makes, models and vehicle types

Yes  No

Other makes and models  Ford Taurus

Yes  No

Other makes and models  Honda Accord

Yes  No

Ford F-150

Taken from Critical Features of High Performance Decision Trees Salford Systems
CART(Growing)

Model

\( (Y_i, (X_i^{(1)}, \ldots, X_i^{(d)})) \in \{1, \ldots, K\} \times \mathbb{R}^d \)

\( i = 1, \ldots, n \)
**CART (Growing)**

\[ \left( \hat{\alpha}_1, \hat{\beta}_1, \hat{\gamma}_1 \right) = \arg\min_{(\alpha_1, \beta_1, \gamma_1) \in \mathbb{R}^3} \sum_{i=1}^{n} \mathbf{1} \left( Y_i \neq \alpha_1 \mathbf{1} \left( X_i^{(1)} \leq \gamma_1 \right) \right) + \mathbf{1} \left( Y_i \neq \beta_1 \mathbf{1} \left( X_i^{(1)} > \gamma_1 \right) \right) \]

\[ \vdots \]

\[ \left( \hat{\alpha}_d, \hat{\beta}_d, \hat{\gamma}_d \right) = \arg\min_{(\alpha_d, \beta_d, \gamma_d) \in \mathbb{R}^3} \sum_{i=1}^{n} \mathbf{1} \left( Y_i \neq \alpha_d \mathbf{1} \left( X_i^{(d)} \leq \gamma_d \right) \right) + \mathbf{1} \left( Y_i \neq \beta_d \mathbf{1} \left( X_i^{(d)} > \gamma_d \right) \right) \]
CART(Growing)

\[
(\hat{\alpha}_1, \hat{\beta}_1, \hat{\gamma}_1) = \arg\min_{(\alpha_1, \beta_1, \gamma_1) \in \mathbb{R}^3} \sum_{i=1}^{n} \left( Y_i \neq \alpha_1 1(X_i^{(1)} \leq \gamma_1) + 1 \left( Y_i \neq \beta_1 1(X_i^{(1)} > \gamma_1) \right) \right)
\]

\[
(\hat{\alpha}_d, \hat{\beta}_d, \hat{\gamma}_d) = \arg\min_{(\alpha_d, \beta_d, \gamma_d) \in \mathbb{R}^3} \sum_{i=1}^{n} \left( Y_i \neq \alpha_d 1(X_i^{(d)} \leq \gamma_d) + 1 \left( Y_i \neq \beta_d 1(X_i^{(d)} > \gamma_d) \right) \right)
\]

\[
\hat{t} = \arg\min_{j=1,\ldots,d} \sum_{i=1}^{n} \left( Y_i \neq \hat{\alpha}_j 1(X_i^{(j)} \leq \hat{\gamma}_j) + 1 \left( Y_i \neq \hat{\beta}_j 1(X_i^{(j)} > \hat{\gamma}_j) \right) \right)
\]
**CART**

**Model**

\[(Y_i, (X_i^{(1)}, \ldots, X_i^{(d)})) \in \{1, \ldots, K\} \times \mathbb{R}^d, \quad i = 1, \ldots, n\]

**CART(Growing)**

\[
(\hat{\alpha}_1, \hat{\beta}_1, \hat{\gamma}_1) = \arg\min_{(\alpha_1, \beta_1, \gamma_1) \in \mathbb{R}^3} \sum_{i=1}^{n} 1(Y_i \neq \alpha_1 1(X_i^{(1)} \leq \gamma_1)) + 1(Y_i \neq \beta_1 1(X_i^{(1)} > \gamma_1))
\]

\[
(\hat{\alpha}_d, \hat{\beta}_d, \hat{\gamma}_d) = \arg\min_{(\alpha_d, \beta_d, \gamma_d) \in \mathbb{R}^3} \sum_{i=1}^{n} 1(Y_i \neq \alpha_d 1(X_i^{(d)} \leq \gamma_d)) + 1(Y_i \neq \beta_d 1(X_i^{(d)} > \gamma_d))
\]

\[\hat{t} = \arg\min_{j=1, \ldots, d} \sum_{i=1}^{n} 1(Y_i \neq \hat{\alpha}_j 1(X_i^{(j)} \leq \hat{\gamma}_j)) + 1(Y_i \neq \hat{\beta}_j 1(X_i^{(j)} > \hat{\gamma}_j))\]
CART(Growing)

\[ \chi(3) \]
CART\((Growing)\)
CART(Growing)
CART (Growing)
CART(pruning)
CART(pruning)
CART(pruning)
CART(pruning)
CART - I

- Advantages
  - Universally applicable to both classification and regression problems.
  - Deals with categorical variables efficiently.
  - Invariant to monotone transformation of input variables.
  - High resistance to irrelevant input variables.
  - Extremely robust to the effect of outliers.
  - Computing is fast.
  - Provide valuable insights for data structure (Interpretation).
**CART - II**

- **Drawbacks**
  - Poor accuracy - SVM often have 30% lower error rates than CART.
  - Instability (high variance) - If we change the data a little, the tree picture can be change a lot.
Example I

- **Internet advertisements data** (From UCI Machine Learning Repository)
  - A set of possible advertisements on internet pages.
  - Task: Predict whether an image is an advertisement.
  - Number of data: 3,279 (458 ads, 2821 non-ads)
  - 1,558 independent variables
    - Geometry of image, phrases occurring in the URL, image’s URL, the anchor text, word near the anchor text.

Accuracy of CART (Matlab): 0.9508 with 10-fold cross validation.
Example II

- Spam E-mail data (From UCI Machine Learning Repository)
  - Task: Classify E-mail as spam or non-spam.
  - Number of data: 4,601 (2788 spam, 1813 non-spam)
  - 57 independent variables
    - Percentage of words in the e-mail that match a certain word.

Accuracy of CART(Matlab): 0.9194 with 10-fold cross validation.
Outline

Motivation

CART

CART construction
Examples

Bagging

Definition
Comparison
Basic Idea and Issues

Random Forests

Definition
Breiman’s Random Forests
Purely Random Forest
Bagging averaged 1-nearest
neighbor classifier
Data Adaptive Weighted Random
Forests

Performances

Example I
Example II
Bagging I

• Ensemble of base learners.

\[
\hat{F}(X) = \begin{cases} 
\frac{1}{M} \sum_{m=1}^{M} T_m(X) & \text{(Regression)} \\
\arg\max_{j} \sum_{m=1}^{M} 1(T_m(X) = j) & \text{(Classification)}
\end{cases}
\]

where \(T_m\): base learner.

• Making base learners is different from Boosting.

• Use bootstrap sample to make base learners.
Bagging II

- Advantages
  - Computing is fast.

- Drawbacks
  - No interpretation.
  - Insufficient analytic results.
Simulation

- \( Y_i = 10 \times X_i + \epsilon_i \)
- \( X_i \sim U(0,1), \epsilon_i \sim N(0,\sigma^2), i = 1, \ldots, n \)
- \( n = 100 \)
- Terminal node size = 5, 20
- \( \sigma = 0.5 \)
Simulation
Simulation

CART vs Bagging (n=100, sigma=0.5, split=5, B=100)

true
CART
Bagging
Loess
**Bias-Variance trade-off I**

If $\mathbb{E} [T_{i,n}] = T_n$ for all $i = 1, \ldots, M$,

$$
\mathbb{E} \left[ \left( \frac{1}{M} \sum_{i=1}^{M} T_{i,n} - \mu \right)^2 \right] = \frac{1}{M^2} \sum_{i=1}^{M} \mathbb{E} \left[ (T_{i,n} - T_n)^2 \right] 
$$

$$
+ \frac{1}{M^2} \sum_{i \neq j} \mathbb{E} \left[ (T_{i,n} - T_n)(T_{j,n} - T_n) \right] 
$$

$$
+ (T_n - \mu)^2
$$

Let $T_{i,n}$ be the $i^{th}$ tree estimator of conditional probability when sample size is $n$ and $\mu = f(x)$, $M$ be the number of trees. (e.g. For original CART, $M = 1$)
Bias-Variance trade-off II

Let $T_{i,n}$ be the $i^{th}$ tree estimators in Bagging. Then, approximately, each $X_i$ uses about $2/3$ of data. Thus, bias of each tree is bigger. But the covariance of $T_{i,n}$ and $T_{j,n}$ is smaller.

(2) $\rightarrow$ smaller

(3) $\rightarrow$ bigger

What if we make (2) much smaller and (3) much larger?
Computation Issue

- If we have $d$ dimensional data set and construct tree to the depth $k$, the total number of computation to choose suitable variable is $d \times (2^{k+1} - 1)$.
- If we randomly choose $F$ variables and use them to select suitable variable at each node, the total number of computation is $M \times F \times (2^{k+1} - 1)$.
- The ratio is $\frac{1}{M} \times \frac{d}{F}$.
- When $F = \lceil \log_2(d + 1) \rceil$ and $M = \sqrt{d}$, the ratio is much less than 1.
- When $d$ is large, computation cost of Random Forests is much cheaper.
Outline

Motivation

CART
- CART construction
- Examples

Bagging
- Definition
- Comparison
- Basic Idea and Issues

Random Forests
- Definition
- Breiman’s Random Forests
- Purely Random Forest
- Bagging averaged 1-nearest neighbor classifier
- Data Adaptive Weighted Random Forests

Performances
- Example I
- Example II
Definition

Random Forests = Random Trees + Aggregation.

• How to make Random Trees (e.g. Random feature selection, Bootstrap sample, Pruning)

• How to assign weight to each tree (e.g. Majority voting, Averaging, Weighted averaging)
Random tree construction

- $Y \in \{-1, 1\}$.
- $\mathbf{X} = (X^{(1)}, \ldots, X^{(10)})$ (i.e. $d = 10$).
- $F = \lceil \log_2(d + 1) \rceil = 3$.
- Generate Bootstrap sample $\mathcal{T}_K$.
- Make maximal tree.
**Single tree construction**

\[(X^{(2)}, X^{(3)}, X^{(8)})\]
Single tree construction

\[ X^{(3)} \rightarrow (X^{(2)}, X^{(3)}, X^{(8)}) \]
Single tree construction

\[ (X^{(2)}, X^{(4)}, X^{(7)}) \rightarrow X^{(7)} \]

\[ X^{(3)} \rightarrow (X^{(2)}, X^{(3)}, X^{(8)}) \]
Single tree construction

$X(3)$

$(X(2), X(3), X(8))$

$(X(4), X(9), X(10))$

$(X(2), X(4), X(7))$

$X(7)$

$(X(4), X(7), X(8))$

$X(3)$
Single tree construction

\[
\begin{align*}
X^{(3)} & \rightarrow (X^{(2)}, X^{(3)}, X^{(8)}) \\
(X^{(2)}, X^{(4)}, X^{(7)}) & \rightarrow X^{(7)} \\
& \rightarrow X^{(4)} \\
& \rightarrow (X^{(4)}, X^{(9)}, X^{(10)})
\end{align*}
\]
Single tree construction

\[
\begin{align*}
X^{(3)} & \quad \rightarrow \quad (X^{(2)}, X^{(3)}, X^{(8)}) \\
(X^{(2)}, X^{(4)}, X^{(7)}) & \quad \rightarrow \quad X^{(7)} & \quad \rightarrow \quad X^{(4)} & \quad \rightarrow \quad (X^{(4)}, X^{(9)}, X^{(10)}) \\
X^{(4)} & \quad \rightarrow \quad \vdots & \quad \rightarrow \quad \vdots & \quad \rightarrow \quad \vdots
\end{align*}
\]
Random Forests construction
Random Forests construction

$k = 1$

$k = 2$

$k = 3$

$k = 4$

$k = 5$

$k = 6$
Algorithm

For $k = 1$ to $M$

(i) Given training training set $\mathcal{T}$, form bootstrap training sets $\mathcal{T}_k$.

(ii) Choose $F$: the number of features.

(iii) At each node in the $k^{th}$ tree, select $F$ features randomly (independently at each node).

(iv) At each node in the $k^{th}$ tree, construct tree-structured classifiers $h(x, \theta_{k_i})$ based on $k_i$ randomly selected features in $\mathcal{T}_k$, where $\theta_{k_i}$ are i.i.d. random vectors.

(v) Grow the tree to maximum depth.
Prediction

For new data $u, v$,

- Calculate the votes or values from each tree.
- Choose majority votes for classification.
- Average the values for regression.
Good properties

- Accuracy is as good as Adaboost and sometimes better.
- Relatively robust to outliers and noise.
- Fast Computation.
- Gives a wealth of important insights (e.g. Estimate of error, variable importance, proximity).
- Simple.
Breiman’s Random Forests

- Breiman (2000), Machine Learning
- Random feature selection
- Maximal trees
- Bootstrap sample
- Majority voting for classification and averaging for regression
Issues about Random Forests

- Why maximal tree?
- Optimal random feature subset size($F$)?
- Bootstrap sample?
- Analytic Results?
Why maximal tree?

- Lin and Jeon (2006), JASA
  - Breiman’s classifier can be viewed as adaptively weighted $k$-potential nearest neighbors methods in regression.
  - Terminal node size should be made to increase with the sample size.
- Biau et al (2008), JMLR
  - Using stopping rule is not necessary in some cases.
- Empirical studies
  - Mark (2004), CBMB: UCI data and simulated data, regression
  - Bae and Bickel (2009), submitted to CSDA: Simulated data, regression and classification
Optimal random features subset size ($F$)

- Many empirical studies
  - Ramón and Sara (2006), BMC Bioinformatics
  - Mark (2004), CBMB
  - Banfield *et al.* (2004), In the Fifth International Conference on Multiple Classifier Systems
  - Bae and Bickel (2009), submitted to CSDA
Bootstrap sample

- Bootstrap sample is not essential for prediction.
- Using bootstrap sample provides useful information.
- But we can get same information by cross validation.
Analytic Results

- Consistency (Biau et al. (2008), JMLR)
  - There exists a distribution of \((X, Y)\) such that \(X\) has non-atomic marginals for which Breiman’s random forest classifier is not consistent.
  - Purely Random Forest
  - Bagging averaged 1-nearest neighbor classifier

- Convergence rate (Bae and Bickel (2009), submitted to JMLR)
  - Data Adaptive Weighted Random Forests
Purely Random Forest (PRF)

- Biau et al (2008), JMLR
- A radically simplified version of random forest classifiers
- At each node, a split variable is selected randomly.
- At each node, a split point is selected according to a uniform random variable on the length of the chosen side of the each.
- Do not use bootstrap sample
- Recursive node splits do not depend on the labels $Y_1, \ldots, Y_n$
Consistency

Consistency of PRF

Assume

- $X$ is supported on $[0, 1]^d$.
- $k \to \infty$ and $\frac{k}{n} \to 0$, where $k$ is the number of nodes, $n$ is the number of data

Then, purely Random Forest classifier is consistent
Bagging averaged 1-nearest neighbor classifier (BNN)

- Biau et al (2008), JMLR
- Generalized version of bagging predictors
- The size of bootstrap sample is not necessary same as the original sample
- Sample without replacement.
- Each data is selected with probability $q_n \in [0, 1]$, independently.
Consistency

Consistency of BNN

The Bagging averaged 1-nearest neighbor classifier is consistent for all distributions of \((X, Y)\) if and only if

- \(q_n \to 0\)
- \(nq_n \to \infty\), \(n\) is the number of data
Data Adaptive Weighted Random Forests (DAWRF)

- Bae and Bickel (2009), submitted to JMLR
- Random Feature selection, BUT same for a tree.
- Do not use bootstrap sample.
- Assign weight to each tree in a data adaptive way.
- Pruning tree
Construction of DAWRF

- For $k = 1$ to $M$
  
  (i) Choose $F_k$(the number of features) randomly from $\{1, \ldots, d\}$.
  
  (ii) Randomly choose a feature subset $S_k$ of $X^{(1)}, \ldots, X^{(d)}$ with size $F_k$
  
  (iii) Construct a classification tree $\hat{f}_k$ using $S_k$ feature variables.
  
  (iv) Compute 1-misclassification error $A(k)$ using another validation data.

- Compute $\hat{W}_k = \frac{\exp(\beta \times A(k))}{\sum_{k=1}^{M} \exp(\beta \times A(k))}$ for suitable $\beta$.

- Define DAWRF classifier as $\sum_{k=1}^{M} \hat{W}_k \hat{f}_k$
**Dyadic Classification Tree (DCT)**

- $L(\phi) = \mathbb{P}[Y \neq \phi(X)]:$ loss function
- $\tilde{L}_n(\phi) = \frac{1}{n} \sum_{i=1}^{n} 1(\phi(X_i) \neq Y_i):$ empirical loss function
- $C^{(k)}$: the collection of all dyadic classification trees with $k$ terminal nodes, $k = 1, \ldots, K$, $K = O\left(n^{(d-1)/d}\right)$.
- $\tilde{\phi}^{(k)}_n = \arg \min_{\phi \in C^{(k)}} \tilde{L}_n(\phi)$

**Dyadic tree classifier:**

\[ \hat{\phi}^*_n = \arg \min_{\tilde{\phi}^{(k)}_n, k = 1, \ldots, K} \tilde{L}_n(\tilde{\phi}^{(k)}_n) + P(k, n), \text{ where} \]

\[ P(k, n) = \lambda \frac{k}{n} (1 + \log d) \text{ is a penalty term for some sufficiently large } \lambda. \]
Bayes decision boundary

- $B(x, \varepsilon)$: the open ball of radius $\varepsilon$ with center $x$
- $\eta(x) = \mathbb{P}(Y = 1 | X = x)$
- $\mathcal{B}$: the Bayes decision boundary

$$\mathcal{B} = \left\{ x \in (0,1)^d : \forall \varepsilon > 0, \exists A_0, A_1 \subset B(x, \varepsilon), \mathbb{P}[A_0] > 0, \mathbb{P}[A_1] > 0, \text{such that } \eta \leq 1/2 \text{ on } A_0, \eta \geq 1/2 \text{ on } A_1 \right\}$$
Assumptions

(C1) \( \eta(x) = \mathbb{P}[Y = 1|X = x] \) is differentiable and
\[ 0 < \delta < \|\eta'(x)\|_\infty < B \]
for \( x \) in the neighborhood of \( \{x : \eta(x) = 1/2\} \).

(C2) (Bounded Marginal): For all sufficiently large \( L \), if we make dyadic cubes with volume \( 2^{-L} \), then for any cube \( A \) intersecting \( B \),
\[ \mathbb{P}[X \in A] \leq C_8 \mu(A) = \frac{C_8}{2^L} \]
where \( \mu \) denotes the Lebesque measure.

(C3) (Regularity): For all sufficiently large \( L \), if we make dyadic cubes with volume \( 2^{-L} \), \( B \) passes through at most \( C_9 2^{L(d-1)/d} \) of the \( 2^L \) cubes.
Theorems

Convergence Rate of DCT

Suppose assumptions (C1),(C2),(C3) satisfy.

Then, there exists a constant $C > 0$ such that

$\mathbb{E} \left[ L(\hat{\phi}_n^*) - L(\phi^*) \right] = \mathbb{E} \left[ L(\hat{\phi}_n^*) - L(\phi^*) \right] \leq C n^{-\frac{1}{d}},$

where $\phi^*(x) = 1$ if $\eta(x) > 1/2$, 0, otherwise.
Theorems

Convergence Rate of DAWRT with DCT

Suppose assumptions (C1),(C2),(C3) satisfy and let \( \hat{\phi}_{n,m} \) be the Data Weighted Random Forests with dyadic classifiers \( \hat{\phi}^*_n \). Then, there exist a constant \( D > 0 \) such that for \( m = O\left(n^{3/2d} \log M\right) \),

\[
\mathbb{E} \left[ L(\hat{\phi}_{n,m}) - L(\phi^*) \right] \leq Dn^{-1/d},
\]

where \( n \) is the number of training sample, \( m \) is the number of validation sample to assign weights and \( M \) is the number of trees.
Remark

• $\hat{\phi}_{n,m}$ is resistant to irrelevant variables.

• When $d^*$ is the dimension of relevant variables, convergence rate is $n^{-1/d^*}$.
Outline

Motivation

CART

CART construction
Examples

Bagging

Definition
Comparison
Basic Idea and Issues

Random Forests

Definition
Breiman’s Random Forests
Purely Random Forest
Bagging averaged 1-nearest neighbor classifier
Data Adaptive Weighted Random Forests

Performances

Example I
Example II
- Number of Trees: 500
- Iteration: 400
- Accuracy estimation: 10 fold cross validation
- Maximal terminal node size for PRF: 20
Example I

Accuracy of Random Forests

Random feature subset size (F)

Accuracy

RF
RF_u
RF_l
CART
DAWRF
PRF
Example I: Effect of terminal node size

![Graph showing the effect of terminal node size on accuracy of Random Forests.](Link to graph)

- RF30
- RF30_u
- RF30_l
- CART
- RF
- Purely RF
- DAWRT

**Motivation**

**CART**

**Bagging**

**Random Forests**

**Performances**
**Example I: Summary**

<table>
<thead>
<tr>
<th></th>
<th>CART</th>
<th>PRF</th>
<th>DAWRF</th>
<th>RF</th>
<th>RF-best</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.9508</td>
<td>0.9643</td>
<td>0.9674</td>
<td>0.9666</td>
<td>0.9724</td>
</tr>
<tr>
<td>sd</td>
<td>0.0112</td>
<td>0.0102</td>
<td>0.0096</td>
<td>0.0114</td>
<td>0.0086</td>
</tr>
<tr>
<td>F</td>
<td>NA</td>
<td>1</td>
<td>NA</td>
<td>10</td>
<td>30</td>
</tr>
</tbody>
</table>
Example II: Performance of BNN
Example II: Effect of random feature size
Example II: Effect of terminal node size
Example II: Summary

<table>
<thead>
<tr>
<th></th>
<th>BNN</th>
<th>CART</th>
<th>DAWRF</th>
<th>PRF</th>
<th>RF</th>
<th>RF-Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.8334</td>
<td>0.9194</td>
<td>0.9378</td>
<td>0.9438</td>
<td>0.9538</td>
<td>0.9565</td>
</tr>
<tr>
<td>sd</td>
<td>0.0178</td>
<td>0.0129</td>
<td>0.0127</td>
<td>0.0106</td>
<td>0.0098</td>
<td>0.0093</td>
</tr>
<tr>
<td>F</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>
Example II: Resistance of irrelevant variables

- Generate 570 irrelevant variables randomly.

<table>
<thead>
<tr>
<th></th>
<th>CART</th>
<th>PRF</th>
<th>DAWRF</th>
<th>RF</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.9103</td>
<td>0.9260</td>
<td>0.9252</td>
<td>0.9453</td>
</tr>
<tr>
<td>sd</td>
<td>0.0138</td>
<td>0.0123</td>
<td>0.0125</td>
<td>0.0096</td>
</tr>
<tr>
<td>F</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>10</td>
</tr>
</tbody>
</table>
Wolpert’s No Free Lunch Theorem

There is no one best algorithm for all problems.

Thank You!