On-line Random Forests

Amir Saffari, Christian Leistner, Jakob Santner
Martin Godec, Horst Bischof

Institute for Computer Graphics and Vision
Graz University of Technology, Austria

October 3, 2009

Motivations

- Random Forest (RF) is an ensemble of random trees.

Motivations

- Random Forest (RF) is an ensemble of random trees.
- RFs achieve state-of-the-art performance in many applications.

Motivations

- Random Forest (RF) is an ensemble of random trees.
- RFs achieve state-of-the-art performance in many applications.
- It is fast both during the the training and testing phase.

Graz University of Technology

Motivations

- Random Forest (RF) is an ensemble of random trees.
- RFs achieve state-of-the-art performance in many applications.
- It is fast both during the the training and testing phase.
- It is easy to implement them in a distributed computing environment or on multi-core CPUs/GPUs.

Motivations

- Random Forest (RF) is an ensemble of random trees.
- RFs achieve state-of-the-art performance in many applications.
- It is fast both during the the training and testing phase.
- It is easy to implement them in a distributed computing environment or on multi-core CPUs/GPUs.
- RFs are inherently multi-class classifiers.

Motivations

- Random Forest (RF) is an ensemble of random trees.
- RFs achieve state-of-the-art performance in many applications.
- It is fast both during the the training and testing phase.
- It is easy to implement them in a distributed computing environment or on multi-core CPUs/GPUs.
- RFs are inherently multi-class classifiers.
- On-line learning is needed for many applications where the size of the data is huge or the data is available from a stream.

Graz University of Technology

Decision Trees

$$\mathcal{X} = \{(x_1, y_1), \cdots, (x_N, y_N)\}, x_i = [x_i^1, \cdots, x_i^D]^T, y_i \in \{1, \cdots, K\}$$

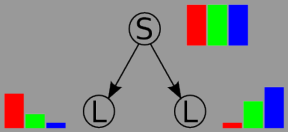
Ⓛ  $p = [p_1, \cdots, p_K]^T$

Ⓢ Split Node

Ⓛ Leaf Node

Decision Trees

$$\mathcal{X} = \{(x_1, y_1), \cdots, (x_N, y_N)\}, x_i = [x_i^1, \cdots, x_i^D]^T, y_i \in \{1, \cdots, K\}$$



$$\text{Test} : g_p(x) > \theta_p, g(x) \in \mathcal{G}$$

$$\text{Gain} : \Delta L = L_j - \frac{|j_r|}{|j|} L_{j_r} - \frac{|j_l|}{|j|} L_{j_l}$$

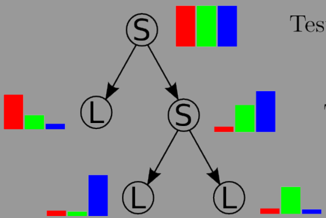
$$\text{Gini index} : L = \sum_{k=1}^{K} p_k (1 - p_k)$$

$$\text{Entropy} : L = -\sum_{k=1}^{K} p_k \log(p_k)$$

$$\text{Feature Test} : \mathcal{G} = \{x^1, \cdots, x^D\}$$

Decision Trees

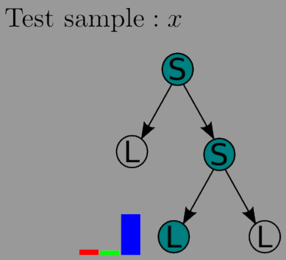
$$\mathcal{X} = \{(x_1, y_1), \cdots, (x_N, y_N)\}, x_i = [x_i^1, \cdots, x_i^D]^T, y_i \in \{1, \cdots, K\}$$



$$\text{Test} : (g_p(x), \theta_p) : x^p > \theta_p$$

$$\text{Test} : (g_r(x), \theta_r) : x^r > \theta_r$$

Decision Trees



$$p(y = k|x) = p_k$$

Decision Trees

- Decision tree is a greedy method which uses a local optimization.

Decision Trees

- Decision tree is a greedy method which uses a local optimization.
- The class of tests could be limited since for finding the best split an optimization step is required.

Decision Trees

- Decision tree is a greedy method which uses a local optimization.
- The class of tests could be limited since for finding the best split an optimization step is required.
- Decision tree is very sensitive to data noise.

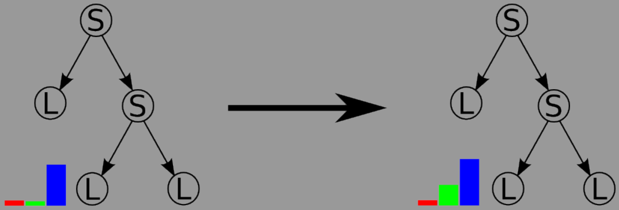
Graz University of Technology

Ensemble of Bagged Trees

L. Breiman (1996)

$$\text{Subsample with replacement} : \mathcal{X} \rightarrow \mathcal{X}_i \cup \mathcal{X}_o$$

$$\text{Train with in-bag-samples} : \mathcal{X}_i \quad \text{Evaluate with out-of-bag-samples} : \mathcal{X}_o$$



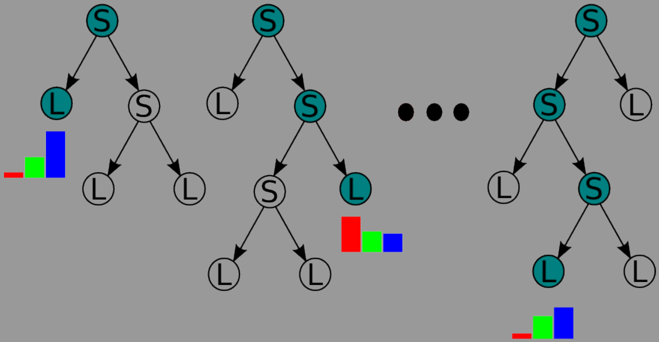
Out-of-bag-error

Refinement

Graz University of Technology

Ensemble of Bagged Trees

Test sample : x



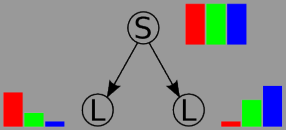
$$p(y = k|x) = \frac{1}{T} \sum_{t=1}^{T} p_t(y = k|x)$$

Graz University of Technology

Random Forests

L. Breiman (2001)

$$\mathcal{X} = \{(x_1, y_1), \cdots, (x_N, y_N)\}, x_i = [x_i^1, \cdots, x_i^D]^T, y_i \in \{1, \cdots, K\}$$



$$\text{Set of Tests} : \mathcal{S} = \{(g_1(x), \theta_1), \cdots, (g_M(x), \theta_M)\}$$

$$\text{Gain} : \Delta L = L_j - \frac{|j_r|}{|j|} L_{j_r} - \frac{|j_l|}{|j|} L_{j_l}$$

$$\text{Feature Test} : \mathcal{G} = \{x^1, \cdots, x^D\}$$

$$\text{Hyperplane Test} : \mathcal{G} = \{g_w(x) = w^T x | w \in R^D\}$$

Graz University of Technology

Elements of On-line Learning

Sample (x, y) is arriving sequentially from a stream.

Graz University of Technology

Elements of On-line Learning

Sample (x, y) is arriving sequentially from a stream.

- On-line bagging.
- On-line random tree growing mechanism.

Graz University of Technology

On-line Bagging

Oza and Russell (2001):

- Draw a random integer: $k \sim \text{Poisson}(\lambda)$

On-line Bagging

Oza and Russell (2001):

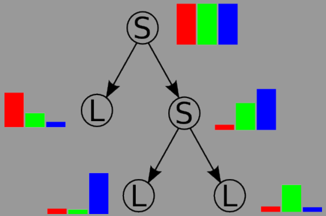
- Draw a random integer: $k \sim \text{Poisson}(\lambda)$
- If $k > 0$:
  - Train the model (tree) on (x, y) k times.
- else:
  - Use (x, y) to compute the out-of-bag-error and refinement.

Graz University of Technology

On-line Random Tree

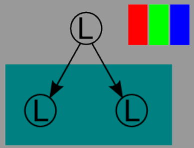
Optimizing the structure of a tree on-line is difficult.

$$(x_i, y_i)$$

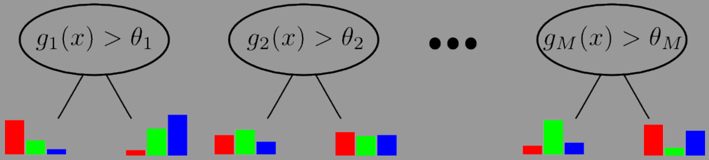


On-line Random Tree

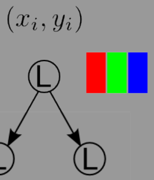
(x_i, y_i)



$$\text{Set of Tests} : \mathcal{S} = \{(g_1(x), \theta_1), \cdots, (g_M(x), \theta_M)\}$$



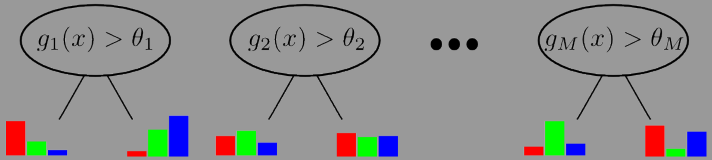
On-line Random Tree



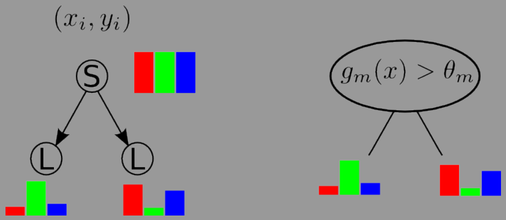
Node size : $|j| > \alpha$

Gain : $\Delta L_m > \beta$

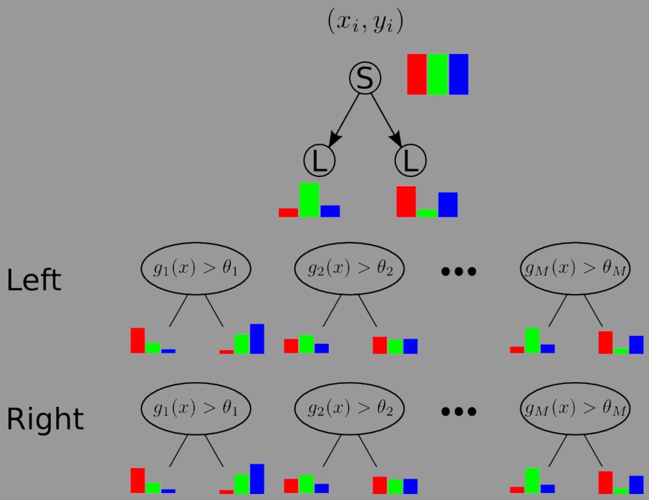
Set of Tests : $\mathcal{S} = \{(g_1(x), \theta_1), \cdots, (g_M(x), \theta_M)\}$



On-line Random Tree



On-line Random Tree



Graz University of Technology

Temporal Knowledge Weighting

- In some applications, the distribution of the data is changing over time.

Temporal Knowledge Weighting

- In some applications, the distribution of the data is changing over time.
- Select a tree randomly from $\{t|t \in \{1, \cdots, T\}, a_t > 1/\gamma\}$.
- If $OOBE_t > \text{rand}()$
  - Discard the t-th tree
  - $f_t = \text{newTree}()$

Machine Learning Datasets

- We set: $T = 200$, $\alpha = 0.1 * N_{train}$, $\beta = 0.1$
- For on-line boosting models, we use 50 selectors with 10 decision stumps in each selector and for multi-class datasets we use a 1-vs-all strategy.
- Code is available at: www.ymer.org/amir/software/online-random-forests

Dataset	# Train	# Test	# Class	# Feat.
Mushrooms	6000x20	2124	2	112
DNA	1400x20	1186	3	180
SatImage	3104x20	2000	6	36
USPS	7291x20	2007	10	256
Letter	15000x20	5000	26	16

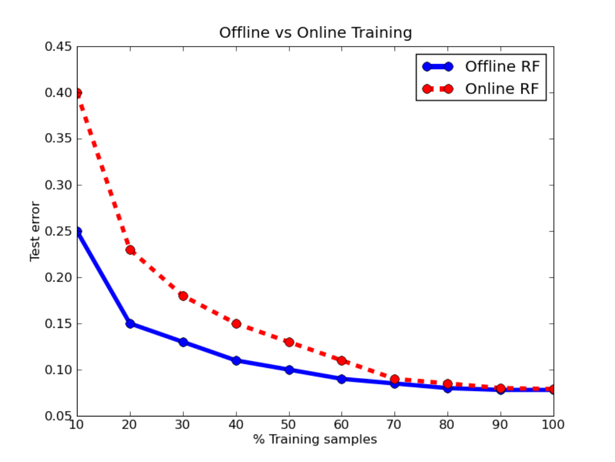
Graz University of Technology

Machine Learning Datasets - Results

Dataset	Off-line RF	On-line RF	On-line Ada	On-line Logit	On-line Savage
Mushrooms	0.010	0.012	0.013	0.012	0.013
DNA	0.109	0.112	0.173	0.117	0.097
SatImage	0.113	0.118	0.257	0.152	0.156
USPS	0.078	0.086	0.224	0.134	0.139
Letter	0.097	0.104	0.263	0.223	0.241

Graz University of Technology

Machine Learning Datasets - Results



TU Graz
Graz University of Technology

Tracking

- We only use simple Haar-features, without implementing any rotation and scale search and avoid any other engineering methods.
- We use 100 trees, $\alpha = 100$, and $\beta = 0.1$.
- For the on-line boosting, we use 50 selectors with each 150 features.
- We evaluate over public datasets: *Occluded Face*, *David Indoor*, *Sylvester*, *Rotating Girl*.
- An implementation of the on-line RF on a common NVidia GPU allows an additional 10-times speed up.

Graz University of Technology

Tracking

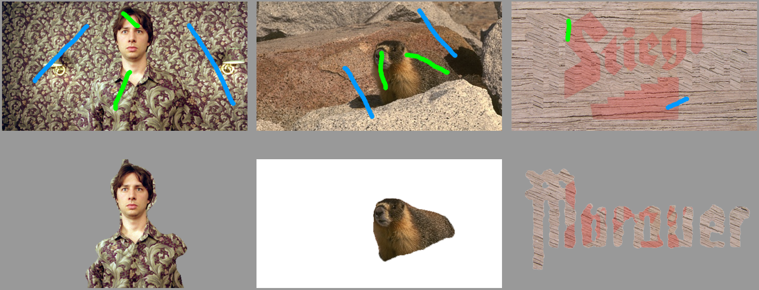
- We only use simple Haar-features, without implementing any rotation and scale search and avoid any other engineering methods.
- We use 100 trees, $\alpha = 100$, and $\beta = 0.1$.
- For the on-line boosting, we use 50 selectors with each 150 features.
- We evaluate over public datasets: *Occluded Face*, *David Indoor*, *Sylvester*, *Rotating Girl*.
- An implementation of the on-line RF on a common NVidia GPU allows an additional 10-times speed up.
- Video

Graz University of Technology

Interactive Segmentation

- We use the interactive segmentation algorithm of Santner et al. (BMVC 2009).
- It uses the off-line RF to learn a foreground model, which then is used as a prior for a weighted Total Variation based segmentation algorithm.
- We replace the off-line RF with our on-line version.
- Both the on-line RF and the segmentation are implemented on a GPU.

Interactive Segmentation



Discussions

Comparison to On-line Boosting

- Robustness to label noise.

Discussions

Comparison to On-line Boosting

- Robustness to label noise.
- Proper plasticity/elacticity trade-off.

Graz University of Technology

Discussions

Comparison to On-line Boosting

- Robustness to label noise.
- Proper plasticity/elacticity trade-off.
- Shrinkage factor effect.

Graz University of Technology

Discussions

Comparison to On-line Boosting

- Robustness to label noise.
- Proper plasticity/elacticity trade-off.
- Shrinkage factor effect.
- Inherently multi-class.

Graz University of Technology

Discussions

Comparison to On-line Boosting

- Robustness to label noise.
- Proper plasticity/elacticity trade-off.
- Shrinkage factor effect.
- Inherently multi-class.
- Suitable for GPU/multi-core/distributed computing.

Graz University of Technology

Thank you!
Code available at:
www.ymer.org/amir/software/online-random-forests

Graz University of Technology