Final Project

Online Random Forests (and where to find them)

🌲 🌳 🌳 🌴 🌲 🌴 🎄 🌳 🎄

Team Overhit

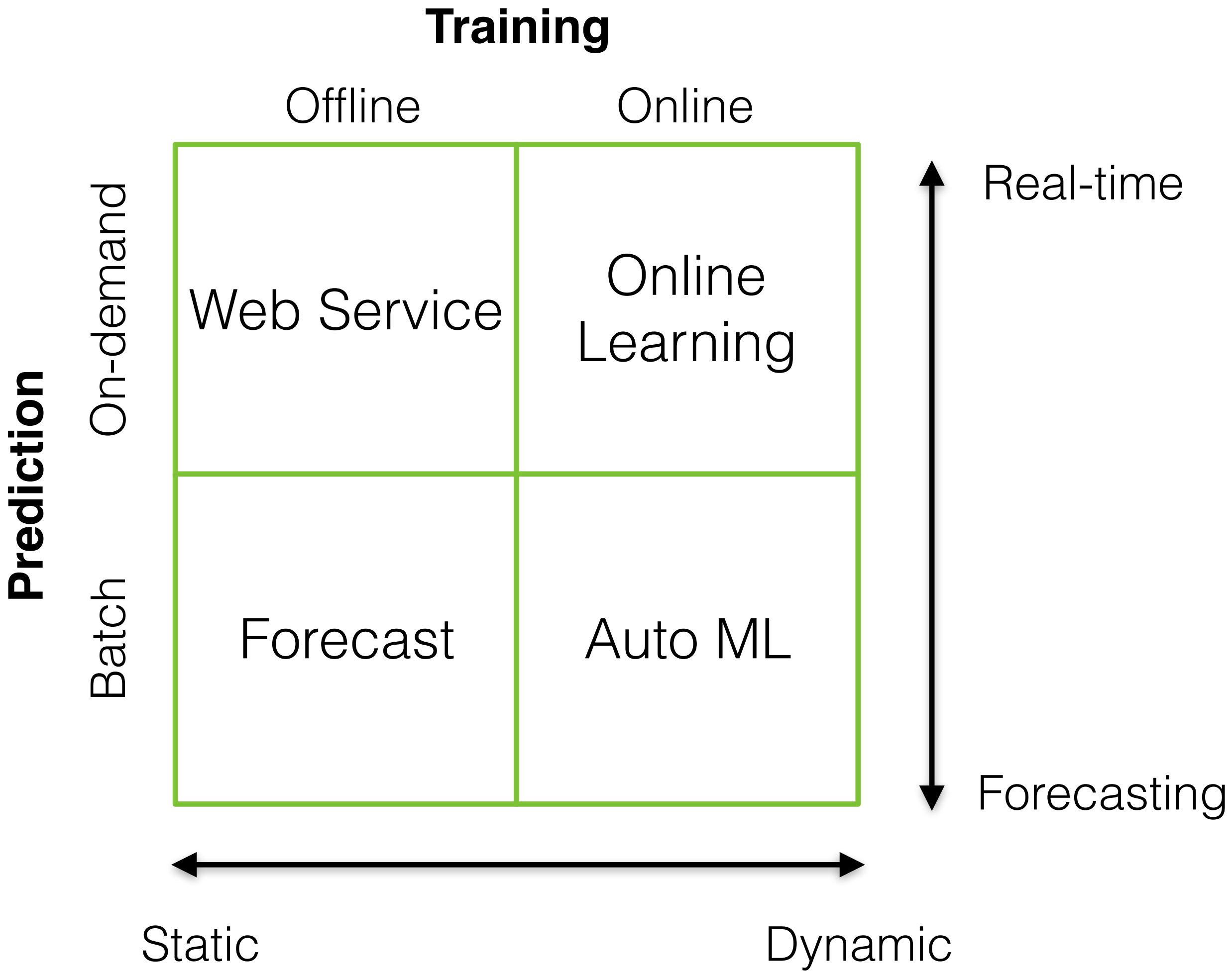
Mikhail Sidorenko

Andrey Demidov

Anton Myshak

Alexey Topolnitsky

What is Online Learning*



Training

Prediction	Offline	Online
On-demand	Web Service	Online Learning
Batch	Forecast	Auto ML

Frameworks for Online Learning

VOWPAL WABBIT

scikit learn

Algorithms

- One-layer NN
- Naive Bayes
- OLS
- Ridge
- Lasso

* https://www.quora.com/How-do-you-take-a-machine-learning-model-to-production

Online Random Forests*

Online Bagging

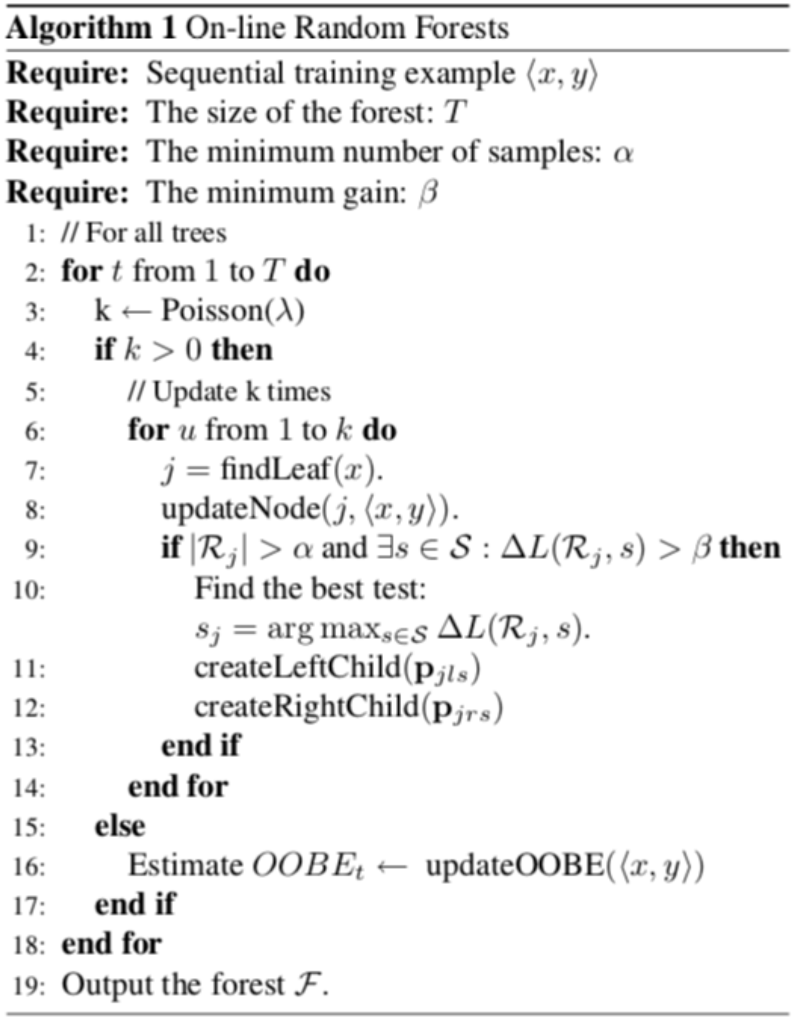
Fit every tree in the ensemble with each new sample **k** times, where **k** is drawn from Poisson (1)

Building trees in online mode

Non-recursive procedure. A node is split only after it saw sufficient amount of samples **alpha** to make statistically significant split

Online adaptation

Randomly delete trees in the ensemble with small OOB error

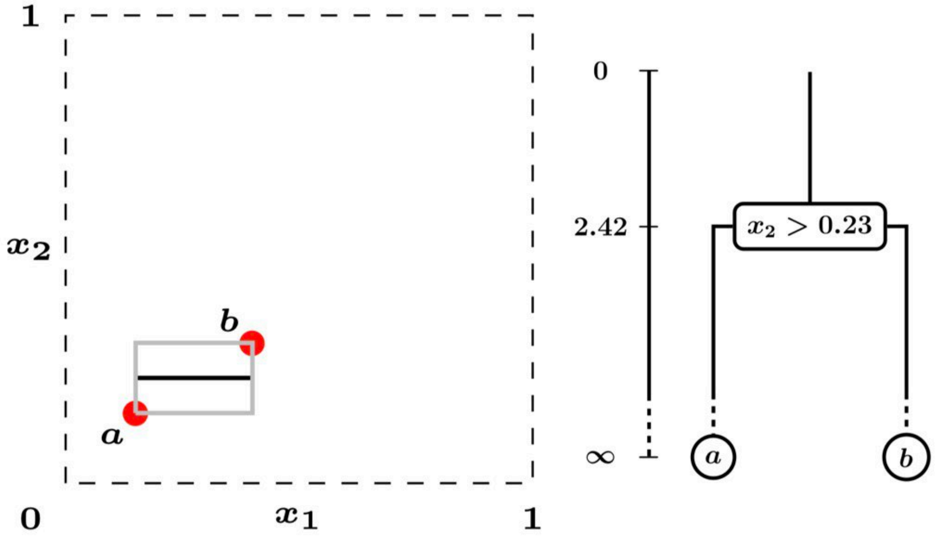


Algorithm 1 On-line Random Forests

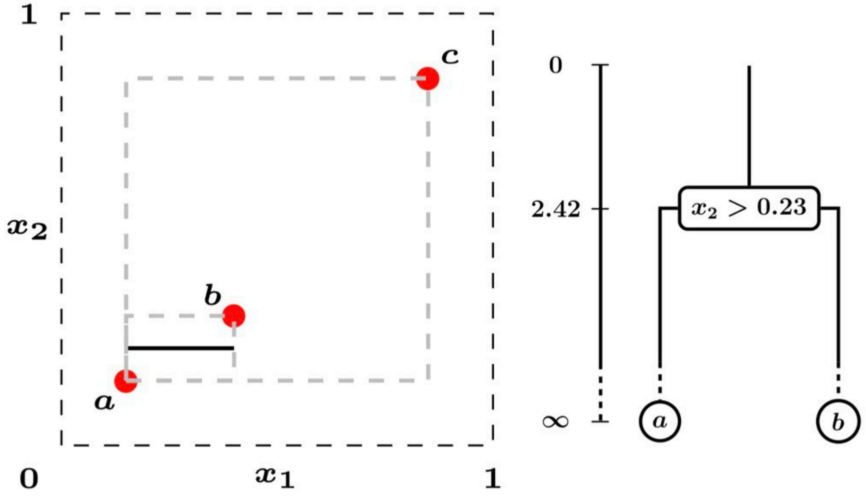
```
Require: Sequential training example $\langle x, y \rangle$
Require: The size of the forest: $T$
Require: The minimum number of samples: $\alpha$
Require: The minimum gain: $\beta$
// For all trees
for $t$ from 1 to $T$ do
    k ← Poisson($\lambda$)
    if $k > 0$ then
        // Update k times
        for $u$ from 1 to $k$ do
            $j$ = findLeaf($x$).
            updateNode($j, \langle x, y \rangle$).
            if $|\mathcal{R}_j| > \alpha$ and $\exists s \in \mathcal{S} : \Delta L(\mathcal{R}_j, s) > \beta$ then
                Find the best test:
                $s_j = \arg\max_{s \in \mathcal{S}} \Delta L(\mathcal{R}_j, s)$.
                createLeftChild($\mathbf{p}_{jls}$)
                createRightChild($\mathbf{p}_{jrs}$)
            end if
        end for
    else
        Estimate $OOBE_t$ ← updateOOBE($\langle x, y \rangle$)
    end if
end for
Output the forest $\mathcal{F}$.
```

* Saffari et al, 2009. «On-line Random Forests»

Mondrian Forests*



Start with data points *a* and *b*

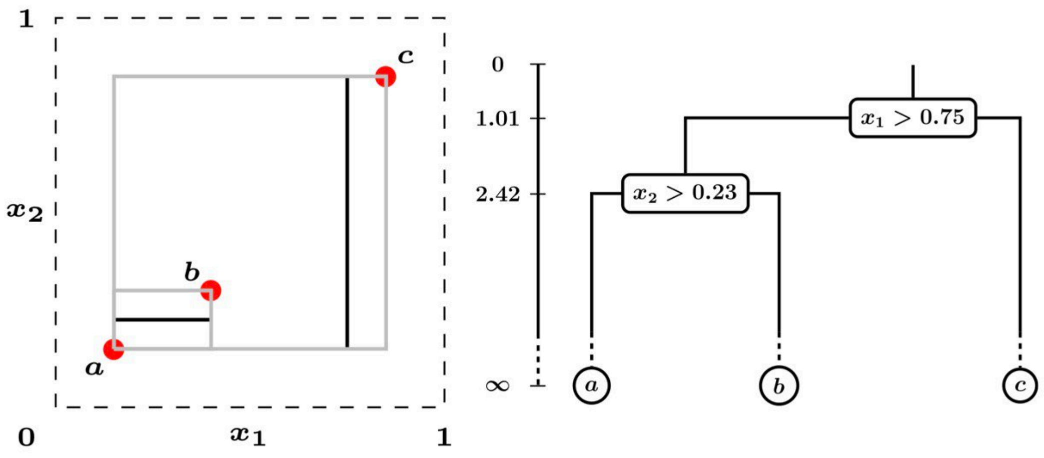


Adding new data point *c*: update range

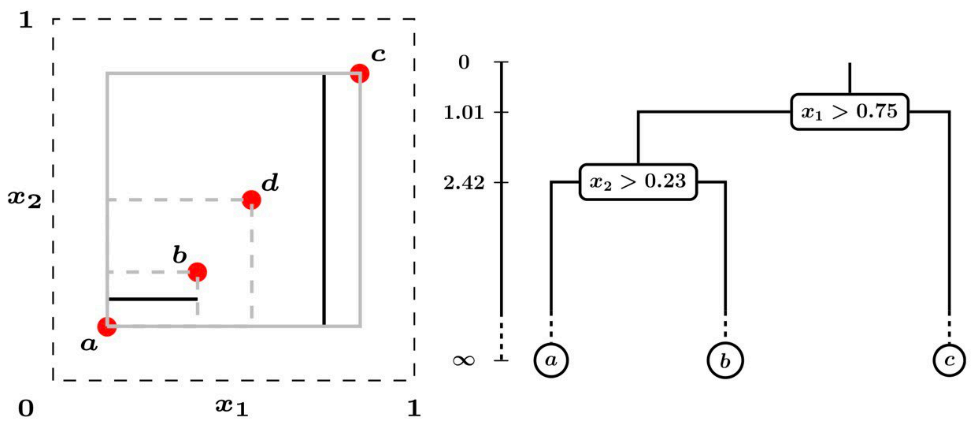
* Lakshiminarayanan et al, 2013. «Mondrian Forests: Efficient Online Random Forests»

Mondrian Forests*

Adding new data point *c*: introduce new split above existing one



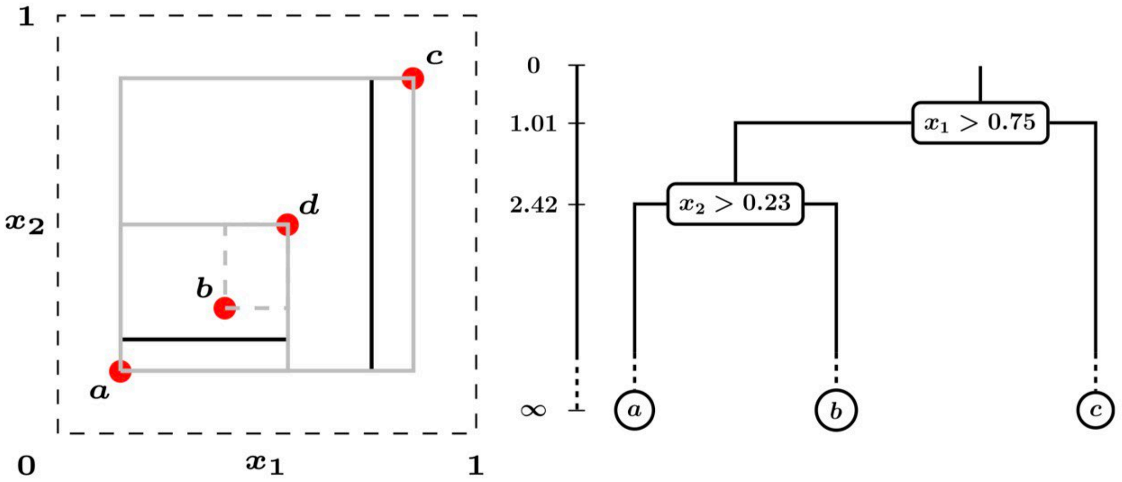
Adding new data point *d*: traverse to left child and update range



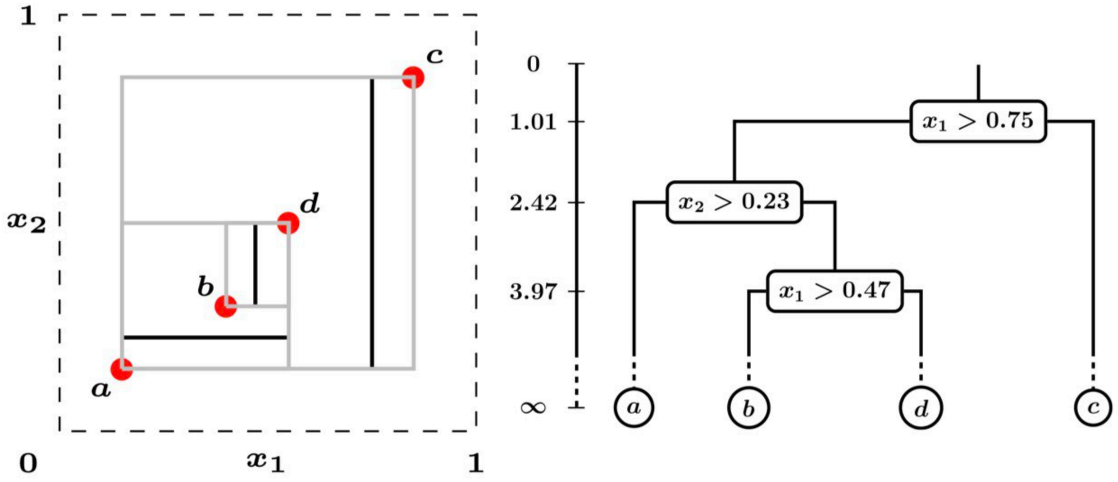
* Lakshiminarayanan et al, 2013. «Mondrian Forests: Efficient Online Random Forests»

Mondrian Forests*

Adding new data point *d*: extend the existing split to new range



Adding new data point *d*: split leaf further



* Lakshiminarayanan et al, 2013. «Mondrian Forests: Efficient Online Random Forests»

Datasets

Handwritten digits USPS dataset

Dataset consists of 16x16 images of digits

Source: https://www.kaggle.com/bistaumanga/usps-dataset

Letter recognition dataset

Dataset consists of parameters of handwritten letters from English alphabet

Source: https://archive.ics.uci.edu/ml/datasets/letter+recognition

Poisson mushrooms recognition dataset

Dataset consists of characteristics of different specious of mushrooms

Source: https://archive.ics.uci.edu/ml/datasets/mushroom

Datasets

Dataset	# Train	# Test	# Class	# Feat
USPS	7291	2007	10	256
Letters	14000	6000	26	16
Mushrooms	5686	2438	2	112

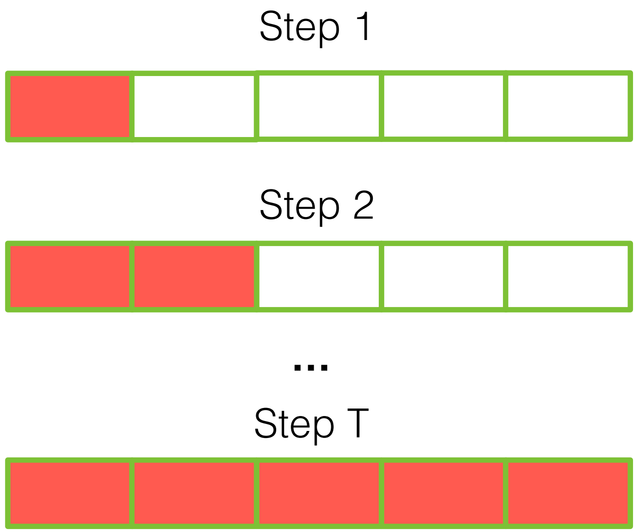
Experimental Setup

Quality Metric: Accuracy

Algorithms: offline Random Forest, online Random Forest, Mondrian Forest
(all algorithms were implemented from scratch and are available in GitHub repo)

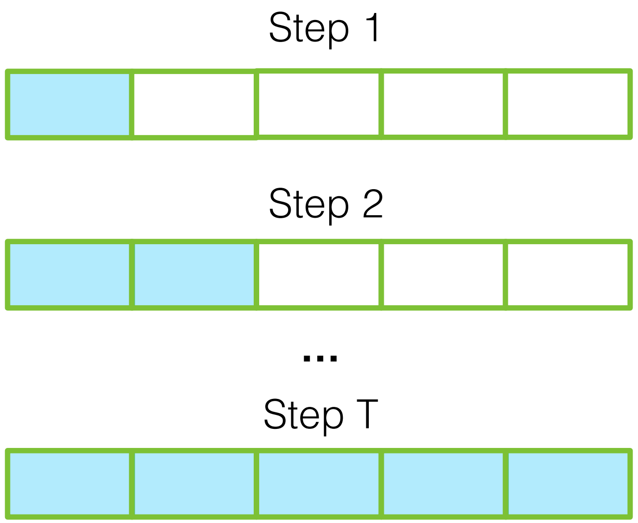
Full Refitting

At each step refit each tree in the ensemble



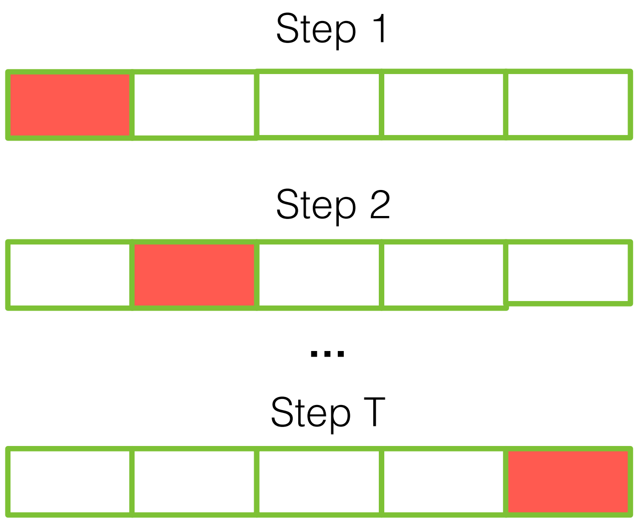
Partial Refitting

At each step refit random set of trees in the ensemble

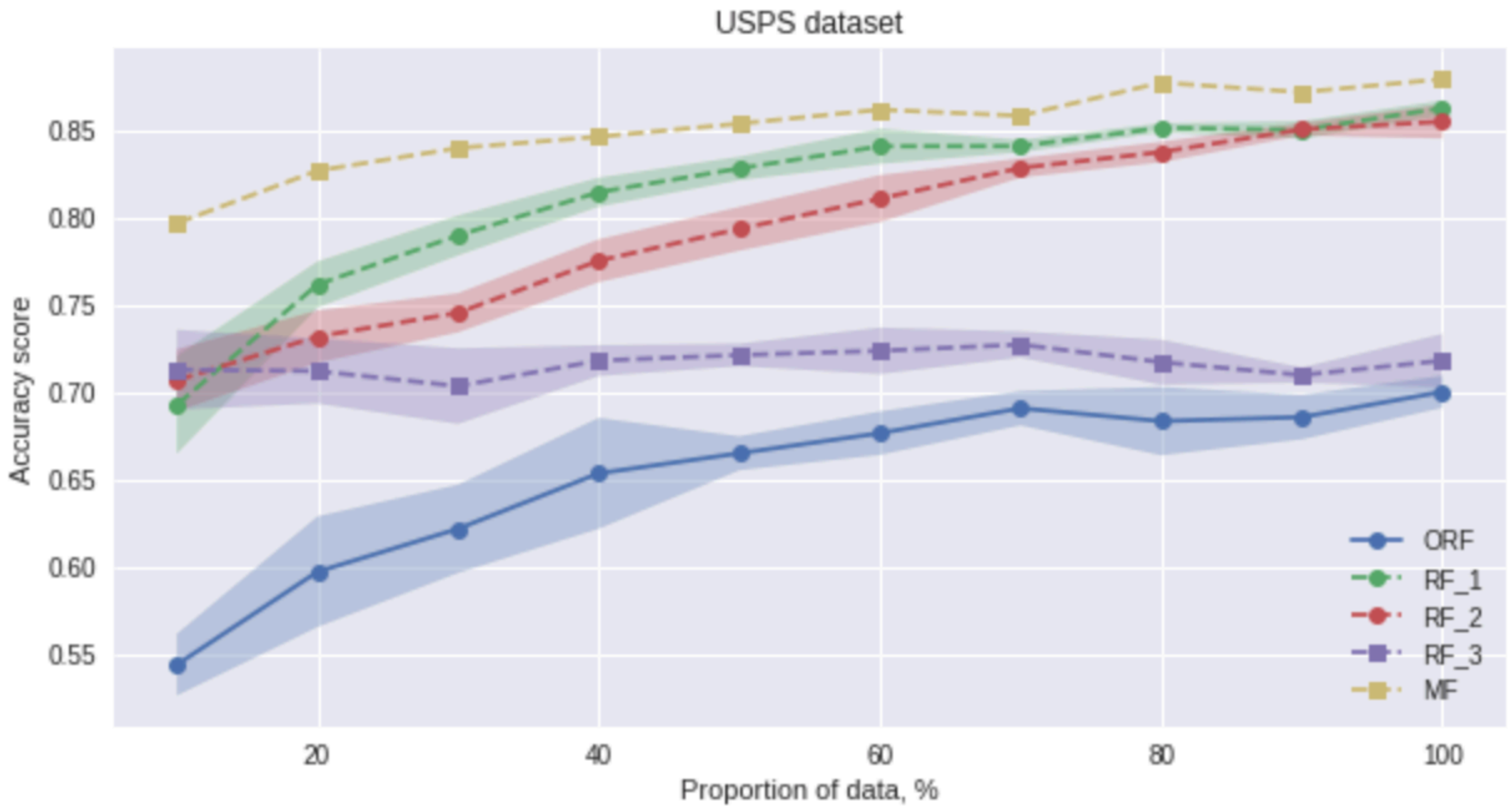


Sliding Window Refitting

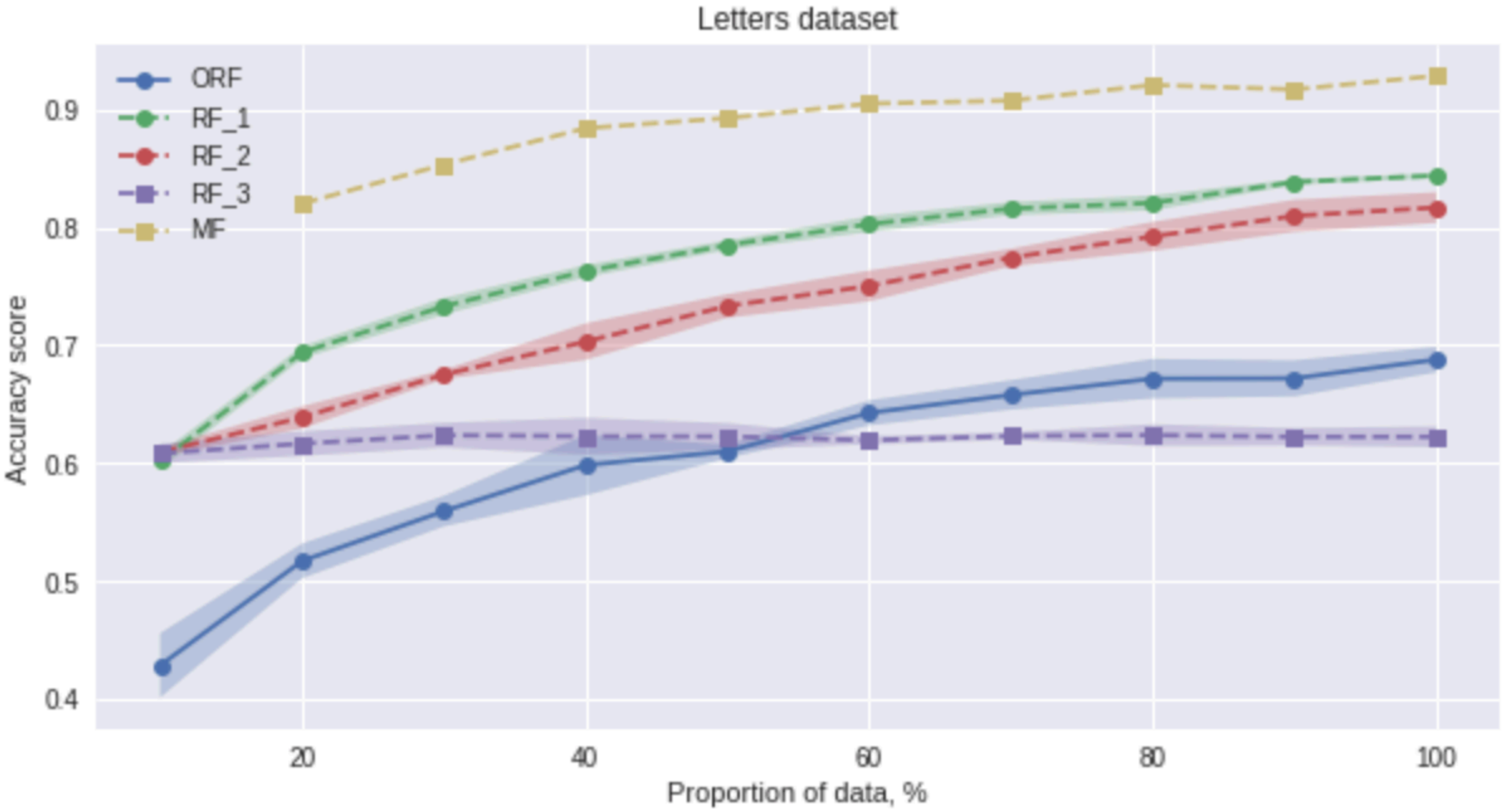
At each step refit random set of trees in the ensemble



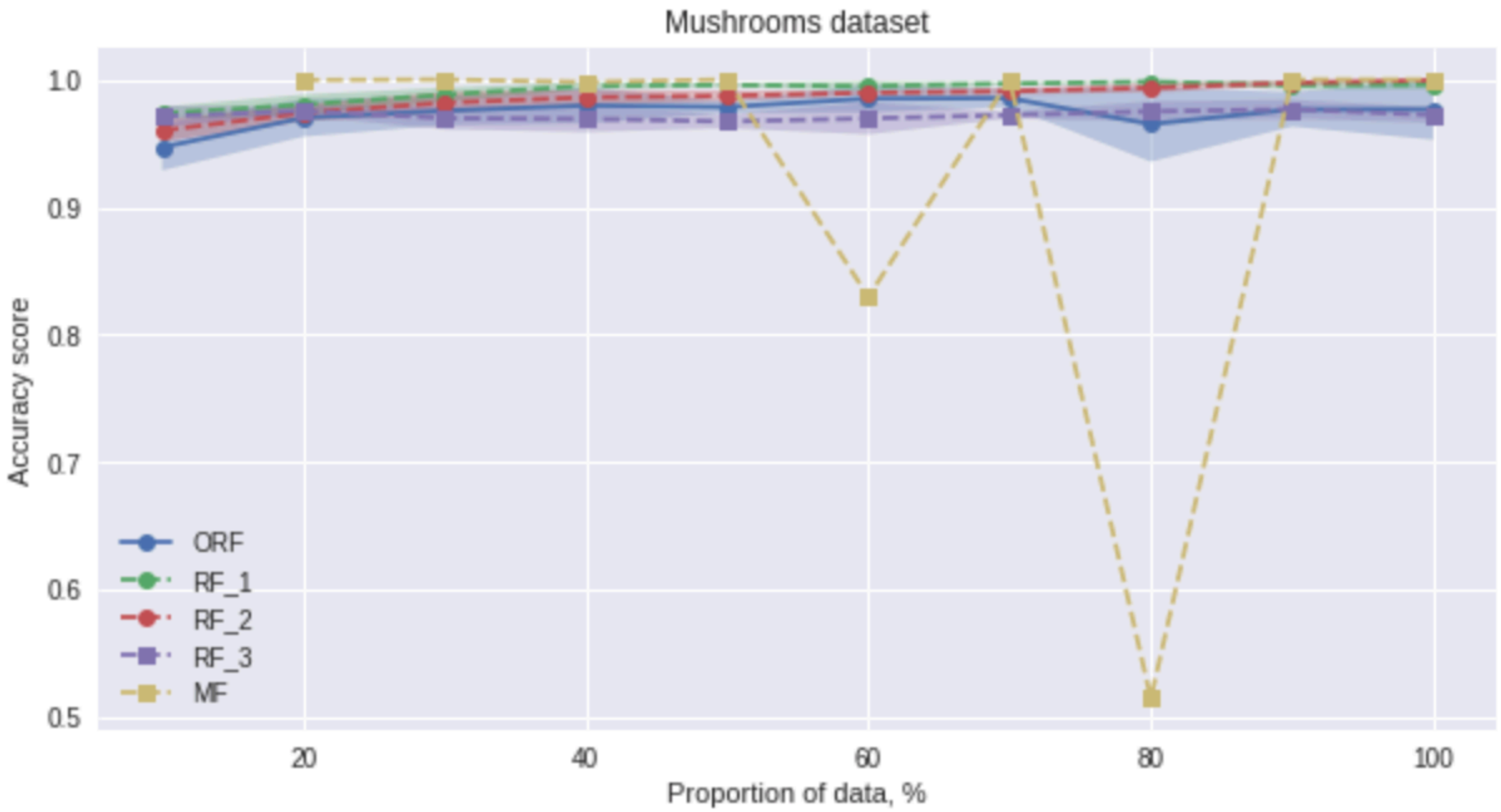
Performance of algorithms



Performance of algorithms



Performance of algorithms



Time complexity of algorithms

	RF-1	RF-2	RF-3	ORF	MRF
USPS	636.11 s	204.36 s	42.53 s	66.6 s	14.1 s
Letters	307.79 s	111.61 s	34.9 s	31.71 s	24.65 s
Mushrooms	32.87 s	14.66 s	9.0 s	3.3 s	2.61 s

1. N. Oza and S. Russell, 2001. Online bagging and boosting. In *Proceedings Artificial Intelligence and Statistics*, pages 105– 112

2. A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1393–1400. IEEE, 2009

3. B. Lakshminarayanan, D.M. Roy, and Teh, Y. W. The. Mondrian Forests: Efficient Online Random Forests. Advances in Neural Information Processing Systems 27, 2014

4. E. Utgoff, N. Bergman, and J. Clouse. Decision tree induc- tion based on efficient tree restructuring. *Machine Learning*, 1997.

Thank you for your attention!

Any questions?