Machine Unlearning for Random Forests

Jonathan Brophy

jbrophy@cs.uoregon.edu

Daniel Lowd

lowd@cs.uoregon.edu

University of Oregon

Motivation

- General Data Protection Regulation (GDPR - 2018)
- Personal Information Protection and Electronic Documents Act (PIPEDA - 2019)
- California Consumer Privacy Act (CCPA - 2020)

“Right to be Forgotten”

Problem Setup

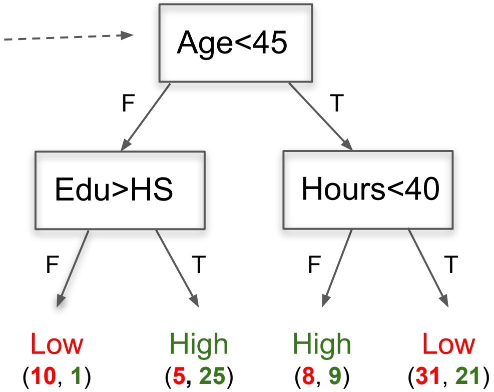
Random Forests:

- Ensemble of decision trees that predicts using majority vote among the trees.
- Each tree trained independently with randomness to increase diversity.

Age	Edu	Hours	Income
50	HS	45	High
32	Col	35	High
65	Grad	30	Low
....			

Decision Tree Learning (in RFs):

- **Interior nodes:** Choose <u>best attribute/value threshold</u>
  (according to split score on training data, from a random set of options).
- **Leaves:** Predict majority class
  (according to examples in training data that match each leaf).



Problem Setup

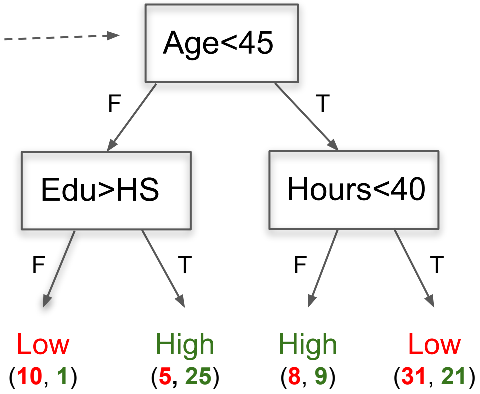
Random Forests:

- Ensemble of decision trees that predicts using majority vote among the trees.
- Each tree trained independently with randomness to increase diversity.

Age	Edu	Hours	Income
50	HS	45	High
~~32~~	~~Col~~	~~35~~	~~High~~
65	Grad	30	Low
....			

Decision Tree Learning (in RFs):

- **Interior nodes:** Choose best attribute/value threshold (according to split score on training data, from a random set of options).
- **Leaves:** Predict majority class (according to examples in training data that match each leaf).



Problem Setup

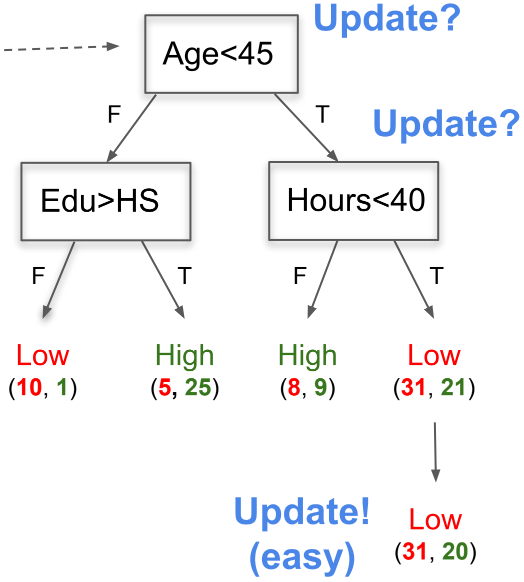
Random Forests:

- Ensemble of decision trees that predicts using majority vote among the trees.
- Each tree trained independently with randomness to increase diversity.

Age	Edu	Hours	Income
50	HS	45	High
~~32~~	~~Col~~	~~35~~	~~High~~
65	Grad	30	Low
....			

Decision Tree Learning (in RFs):

- **Interior nodes:** Choose <u>best attribute/value threshold</u> (according to split score on training data, from a random set of options).
- **Leaves:** Predict majority class (according to examples in training data that match each leaf).



Problem Setup

Random Forests:

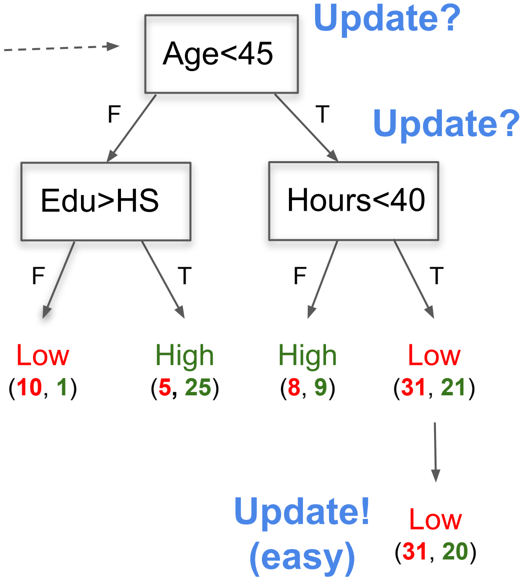
- Ensemble of decision trees that predicts using majority vote among the trees.
- Each tree trained independently with randomness to increase diversity.

Decision Tree Learning (in RFs):

- **Interior nodes:** Choose <u>best attribute/value threshold</u> (according to split score on training data, from a random set of options).
- **Leaves:** Predict majority class (according to examples in training data that match each leaf).

Naive approach:
Retrain from scratch on an updated dataset.

Age	Edu	Hours	Income
50	HS	45	High
~~32~~	~~Col~~	~~35~~	~~High~~
65	Grad	30	Low
....			



Problem Setup

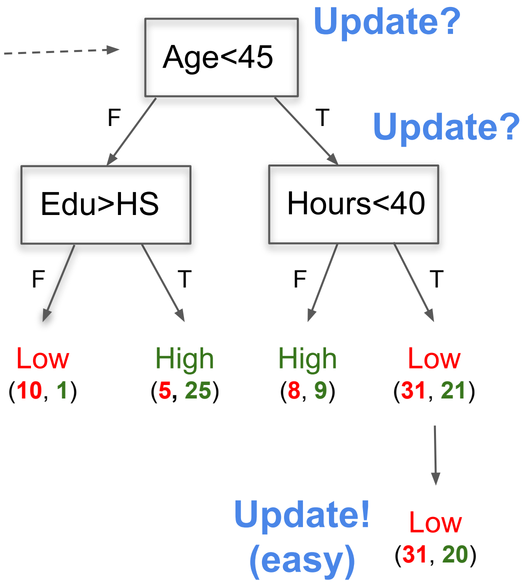
Random Forests:

- Ensemble of decision trees that predicts using majority vote among the trees.
- Each tree trained independently with randomness to increase diversity.

Age	Edu	Hours	Income
50	HS	45	High
~~32~~	~~Col~~	~~35~~	~~High~~
65	Grad	30	Low
....			

Decision Tree Learning (in RFs):

- **Interior nodes:** Choose best attribute/value threshold
  (according to split score on training data, from a random set of options).
- **Leaves:** Predict majority class
  (according to examples in training data that match each leaf).



Naive approach:
Retrain from scratch on an updated dataset.

Data Removal-Enabled RFs (DaRE RFs) (our approach):
Through careful caching and randomness, we can perform updates ***ORDERS OF MAGNITUDE FASTER*** than retraining.

Goal

Efficiently remove the *effect* of a training instance from the specified model.

$$r = \frac{P(R(A(D), D, (x,y)))}{P(A(D \backslash (x,y)))}^{*}$$

D: Dataset
A: Randomized learning algorithm
R: Removal mechanism
(x, y): Instance to remove

$r = 1$: Exact unlearning (a.k.a. perfect unlearning)

$r \cong 1$: Approximate unlearning (a.k.a. statistical unlearning)

*Based on def. (1) from Guo et al. (2019), which was inspired by differential privacy.

Goal

Efficiently remove the *effect* of a training instance from the specified model.

$$r = \frac{P(R(A(D),D,(x,y)))}{P(A(D \setminus (x,y)))}$$

D: Dataset
A: Randomized learning algorithm
R: Removal mechanism
(x, y): Instance to remove

$\boxed{r = 1\text{: Exact unlearning (a.k.a. perfect unlearning)}}$ **This work!**

Theorem 3.1. *Data deletion for DaRE forests is exact, meaning that removing instances from a DaRE model yields exactly the same model as retraining from scratch on updated data.*

Our Contribution: DaRE RFs

[Recap] Decision Tree Learning (in RFs):

- **Interior nodes:** Choose best attribute/value threshold
  (according to split score on training data, from a random set of options).

- **Leaves:** Predict majority class
  (according to examples in training data that match each leaf).

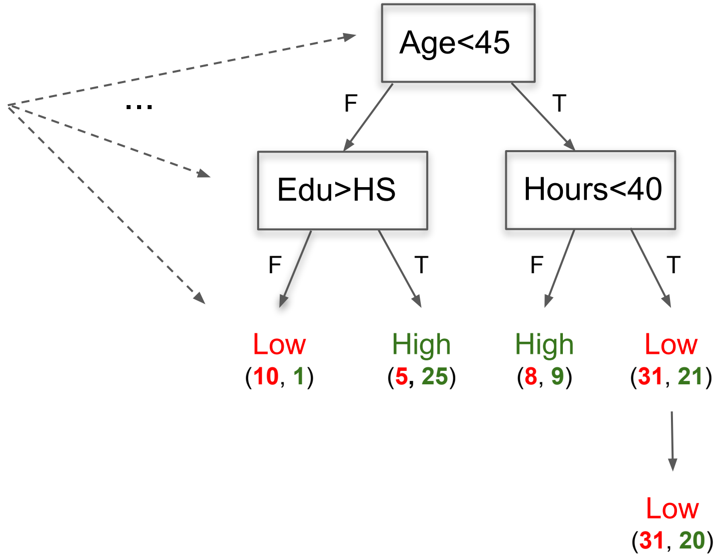
Age	Edu	Hours	Income
50	HS	45	High
~~32~~	~~Col~~	~~35~~	~~High~~
65	Grad	30	Low
....			

DaRE RFs:

1. **Cache statistics to avoid rescoring all splits. Only retrain subtree when the best split changes.**

2. Only consider a random subset of thresholds to keep statistics compact.

3. Use some random split nodes that minimally depend on the data and thus rarely need to be retrained.



Our Contribution: DaRE RFs

[Recap] Decision Tree Learning (in RFs):

- **Interior nodes:** Choose <u>best attribute/value threshold</u> (according to split score on training data, from a random set of options).

- **Leaves:** Predict majority class (according to examples in training data that match each leaf).

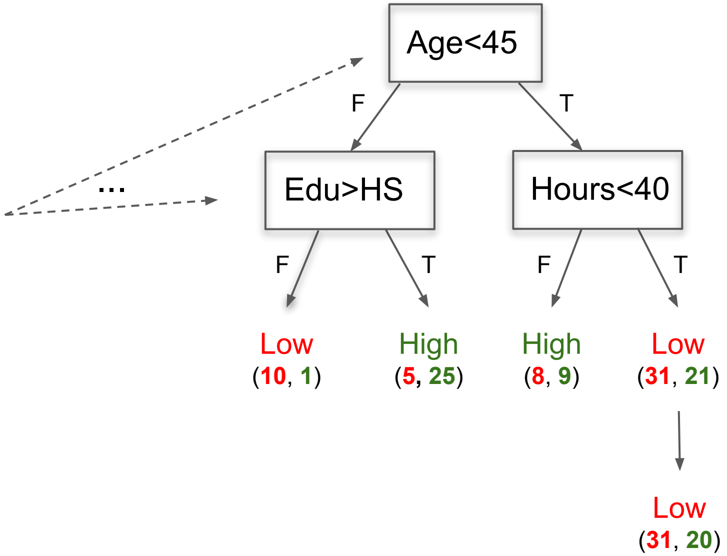
Age	Edu	Hours	Income
50	HS	45	High
~~32~~	~~Col~~	~~35~~	~~High~~
65	Grad	30	Low
....			

DaRE RFs:

1. <u>Cache statistics</u> to avoid rescoring all splits. Only retrain subtree when the best split changes.

2. **Only consider a random <u>subset of thresholds</u> to keep statistics compact.**

3. Use some <u>random split nodes</u> that minimally depend on the data and thus rarely need to be retrained.



Our Contribution: DaRE RFs

[Recap] Decision Tree Learning (in RFs):

- **Interior nodes:** Choose <u>best attribute/value threshold</u>
  (according to split score on training data, from a random set of options).

- **Leaves:** Predict majority class
  (according to examples in training data that match each leaf).

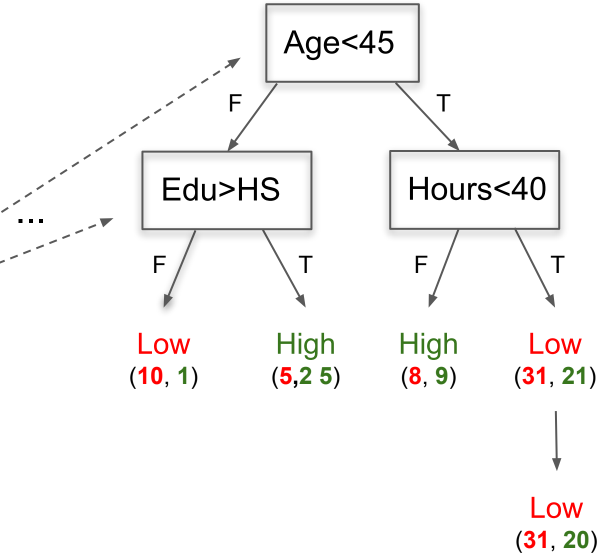
Age	Edu	Hours	Income
50	HS	45	High
~~32~~	~~Col~~	~~35~~	~~High~~
65	Grad	30	Low
....			

DaRE RFs:

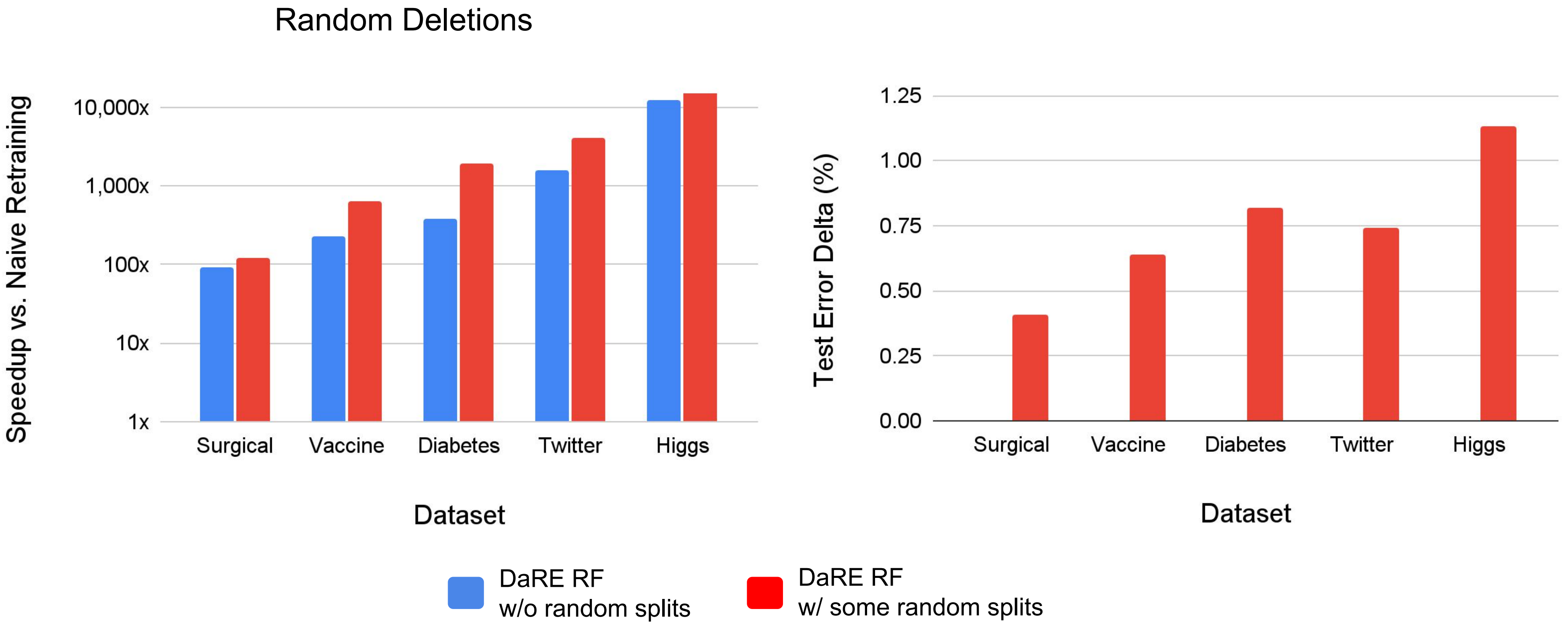
1. <u>Cache statistics</u> to avoid rescoring all splits.
   Only retrain subtree when the best split changes.

2. Only consider a random <u>subset of thresholds</u> to keep statistics compact.

3. **Use some <u>random split nodes</u> that minimally depend on the data and thus rarely need to be retrained.**

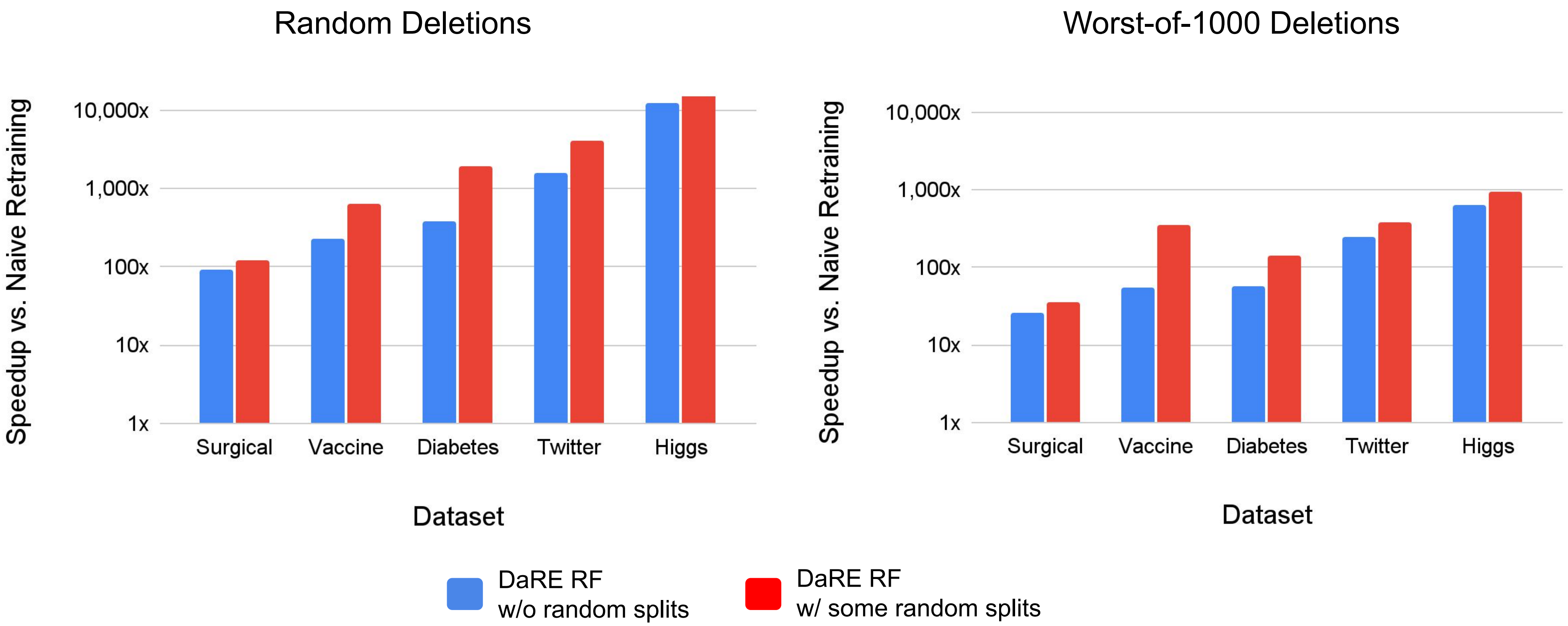


Deletion Efficiency



Theorem 3.3. *Deleting an instance from a DaRE tree requires time* $O(p\ k\ d_{max})$*, where* d_{max} *is the maximum depth of the tree, and p and k are the number of attributes and thresholds to consider at each decision node, respectively.*

Deletion Efficiency



Space Overhead

Memory usage and Overhead. Overhead = (data + DaRE RF) / (data + SKLearn RF).

Dataset	Data	DaRE RF	SKLearn RF	Overhead
Surgical	4 MB	390 MB	30 MB	12x
Vaccine	20 MB	430 MB	40 MB	8x
Diabetes	80 MB	5,000 MB	260 MB	15x
Twitter	50 MB	2,500 MB	330 MB	8x
Higgs	1,000 MB	39,000 MB	1,300 MB	17x

Theorem 3.4. *The space complexity of a DaRE forest is* $O(k\,p\,2^{d_{max}}\,T + n\,T)$*, where* n *is the number of training examples and* T *is the number of trees.*

Take-Home Message

DaRE RFs:

- Delete data orders of magnitude faster than retraining from scratch while sacrificing little to no predictive performance.

- Introduce a trade-off between deletion efficiency and space overhead; overall, 10-20x larger model means we get 100-10,000x faster deletions.

github.com/jjbrophy47/dare_rf