

Ensemble Learning at Scale

Software, Hardware and Algorithmic Approaches



Erin LeDell Ph.D.
Machine Learning Scientist
H2O.ai

Introduction

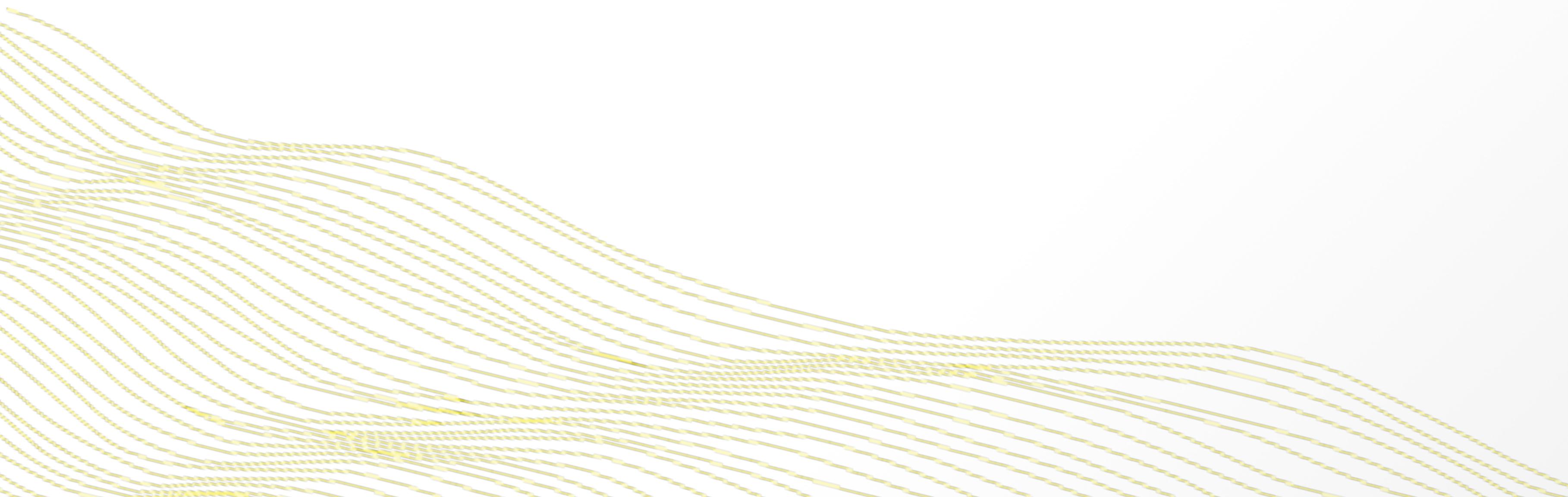
- Statistician & Machine Learning Scientist at H2O.ai in Mountain View, California, USA
- Ph.D. in Biostatistics with Designated Emphasis in Computational Science and Engineering from UC Berkeley (focus on Machine Learning)
- Worked as a data scientist at several startups
- Develop open source machine learning software

Agenda



- Introduction
- Motivation
- Super Learner Algorithm / Stacking
- Subsemble Algorithm
- Online Super Learning
- H2O Ensemble

Motivation



Common Types of Ensemble Methods

Bagging

- Reduces variance and increases accuracy
 - Robust against outliers or noisy data
 - Often used with Decision Trees (i.e. Random Forest)
-

Boosting

- Also reduces variance and increases accuracy
 - Not robust against outliers or noisy data
 - Flexible – can be used with any loss function
-

Stacking

- Used to ensemble a diverse group of strong learners
- Involves training a second-level machine learning algorithm called a “metalearner” to learn the optimal combination of the base learners

Why Ensembles?

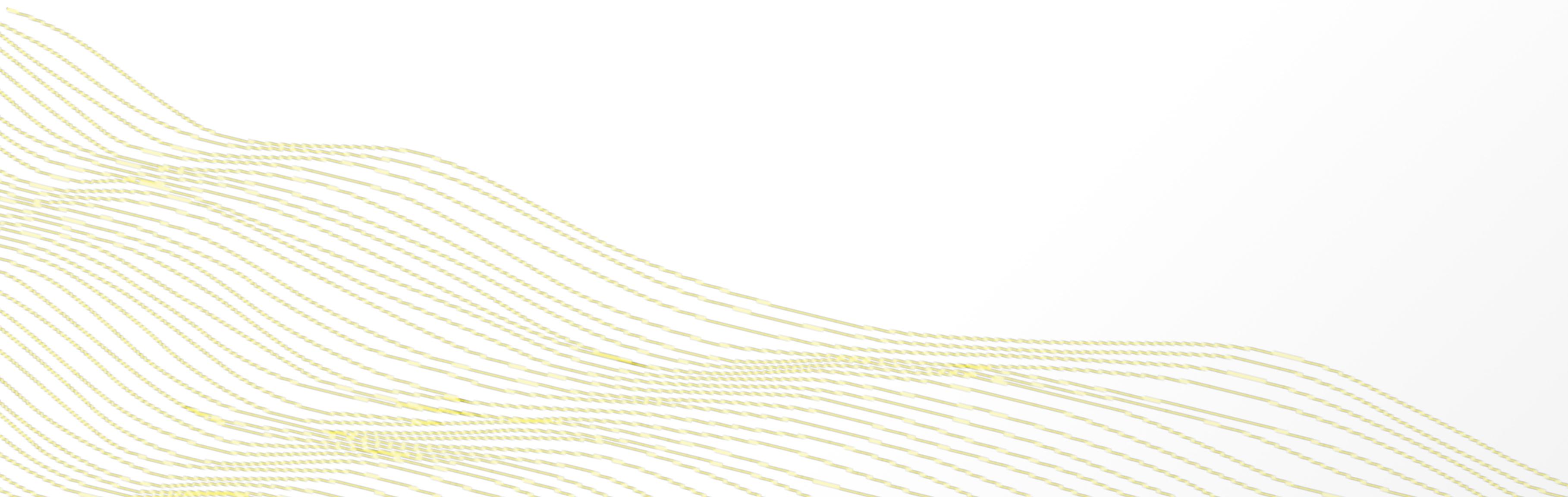
Strength

- A multi-algorithm ensemble may better approximate the true prediction function better than a single algorithm.
- If model performance is the #1 priority, then multi-algorithm ensembles are a fantastic choice. This is evidenced by the fact that Stacking wins nearly all Kaggle competitions.

Weakness

- Increased training & prediction times.
- Increased model complexity.
- Requires large machines or clusters for big data.
- Lack of comprehensive software featuring a large number of algorithms with a single API.

Super Learner / Stacking



History of Stacking

Stacked Generalization

- David H. Wolpert, "Stacked Generalization" (1992)
 - First formulation of stacking via a metalearner
 - Blended Neural Networks
-

Stacked Regressions

- Leo Breiman, "Stacked Regressions" (1996)
 - Modified algorithm to use CV to generate level-one data
 - Blended Neural Networks and GLMs (separately)
-

Super Learning

- Mark van der Laan et al., "Super Learner" (2007)
- Provided the theory to prove that the Super Learner is the asymptotically optimal combination
- First R implementation in 2010; "SuperLearner" package

The Super Learner Algorithm



The Super Learner Algorithm

$$n \left\{ \overbrace{\begin{bmatrix} x \end{bmatrix}}^m \right] \begin{bmatrix} y \end{bmatrix}$$

“Level-zero”
data

- Start with design matrix, X , and response column, y .
- Specify L base learners (with model parameters).
- Specify a metalearner (just another algorithm).
- Perform k-fold CV on each of the L learners.

The Super Learner Algorithm

$$n \left\{ \begin{bmatrix} p_1 \\ \vdots \\ p_L \end{bmatrix} \cdots \begin{bmatrix} p_1 \\ \vdots \\ p_L \end{bmatrix} \begin{bmatrix} y \end{bmatrix} \right\} \rightarrow n \left\{ \underbrace{\begin{bmatrix} z \\ \vdots \\ z \end{bmatrix}}_{L} \begin{bmatrix} y \end{bmatrix} \right\}$$

“Level-one”
data

- Collect the predicted values from k-fold CV that was performed on each of the L base learners.
- Column-bind (“stack”) these prediction vectors together to form a new design matrix, Z.
- Train the metalearner using Z, y.

Scalable Stacking Solutions

The original SuperLearner R package works well for small datasets, 😊 but not for medium or large datasets. 😞

- Develop alternative formulations of Super Learner than can learn on subsets of the data.
- Use a more scalable language (C++, Java, Scala, Julia) to implement Super Learner.
- Use algorithms that learn iteratively and thus do not need to load the entire training set into memory at once (aka. online learning).

Scalable Stacking Solutions

Subsemble

- An R package that implements the Subsemble algorithm for combining models trained on partitions of the data, a variant of Super Learning.
 - Like SuperLearner, can be used with any R algorithm.
-

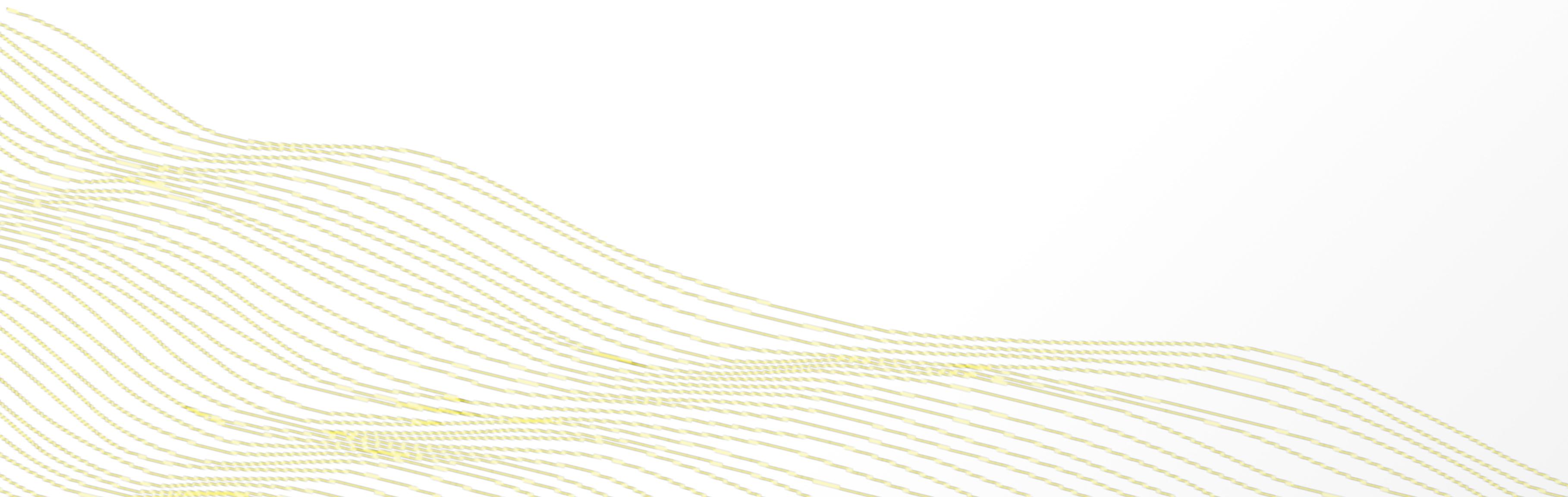
Online SuperLearner

- Implements an online version of the Super Learner algo.
 - Built on top of the out-of-core, online machine learning software, Vowpal Wabbit (VW), in C++.
-

H2O Ensemble

- H2O Ensemble implements the standard Super Learner algorithm using H2O distributed algorithms.
- No limit to the size of the cluster that can be used, so it's ideal for datasets bigger than a single node's RAM.

Subsemble



The Subsemble Algorithm

- Subsemble is a general subset ensemble prediction method which can be used for small, moderate, or large datasets.
- Subsemble partitions the full dataset into subsets of observations, fits a specified underlying algorithm on each subset, and uses a unique form of k-fold cross-validation to output a prediction function that combines the subset-specific fits.
- An oracle result provides a theoretical performance guarantee for Subsemble.

The Subsemble Algorithm

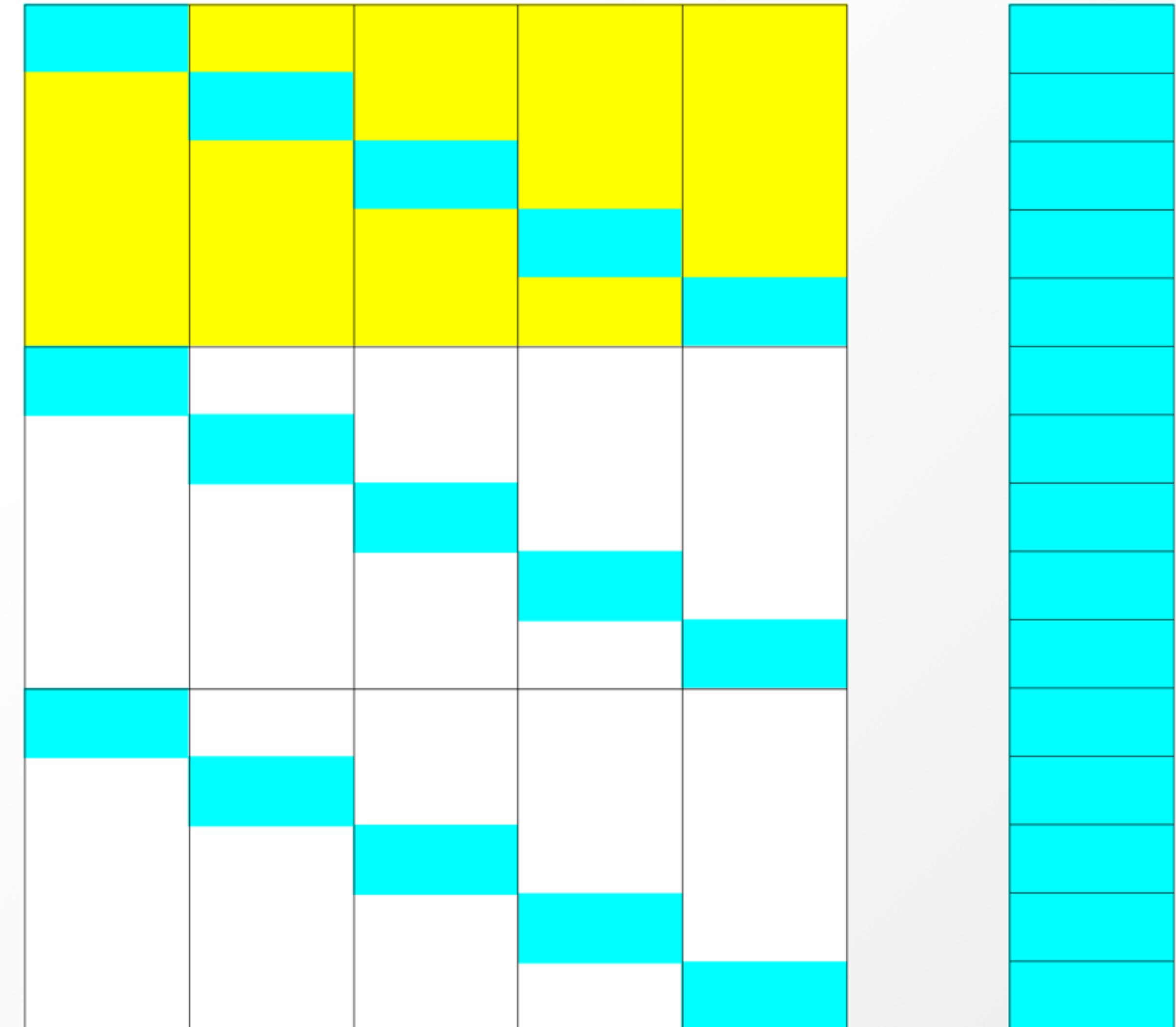
$$n \left\{ \begin{bmatrix} m \\ x \end{bmatrix} \right| y \right\}$$

“Level-zero”
data

- Start with design matrix, X , and response column, y .
- Specify L base learners and a metalearner.
- Partition the data into J subsets.
- Partition each subset into V folds.
- Note: The v^{th} validation fold spans all J subsets.

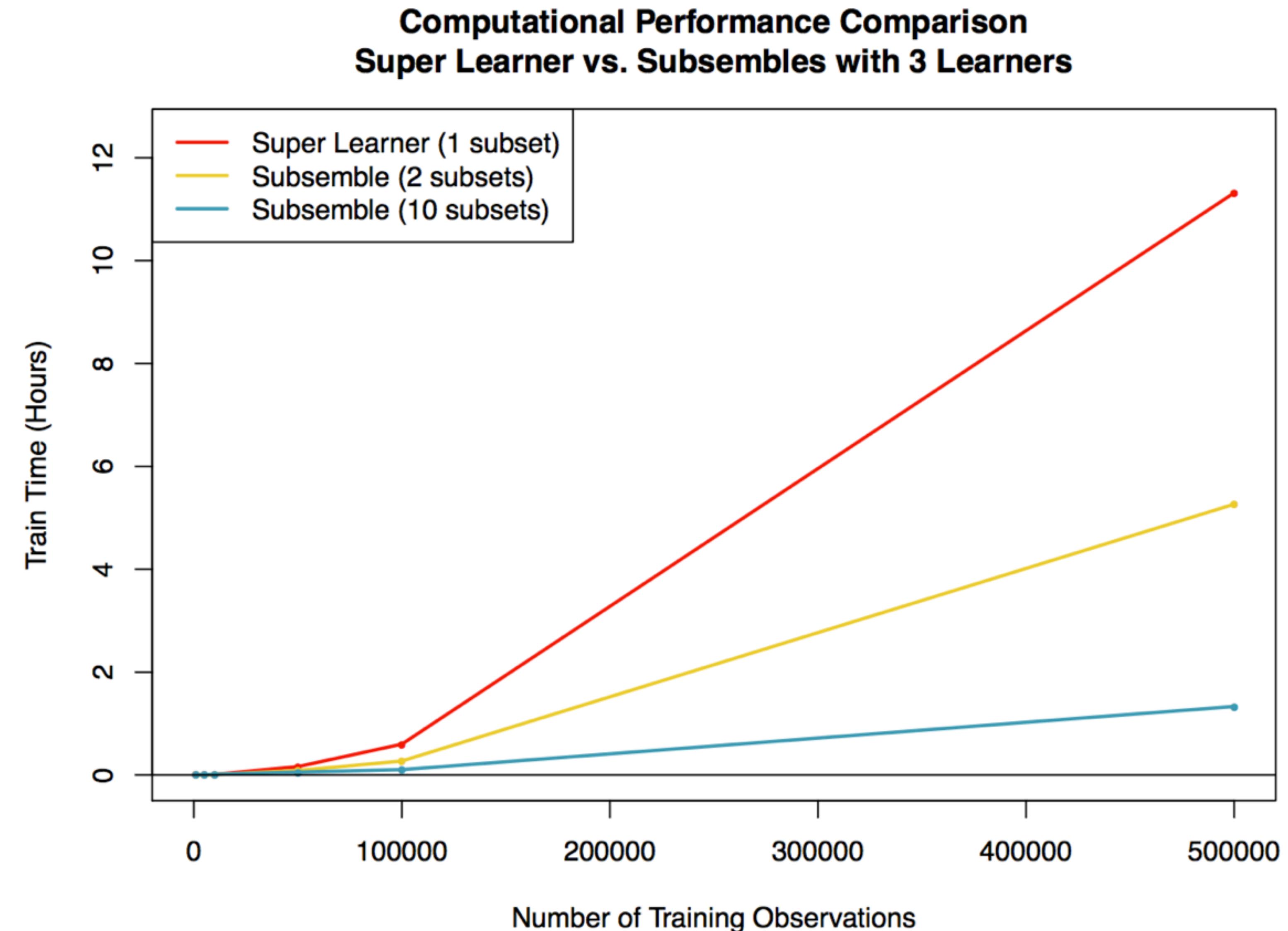
Modified K-fold CV in Subsemble

- An example of 5-fold cross-validation for the first subset with a single base learner.
- Even though Subsemble trains models on subsets of the data, it still produces predictions for all n rows.
- Instead of producing an $n \times L$ level-one matrix (as in Super Learner), Subsemble produces an $n \times (J \times L)$ level-one matrix.



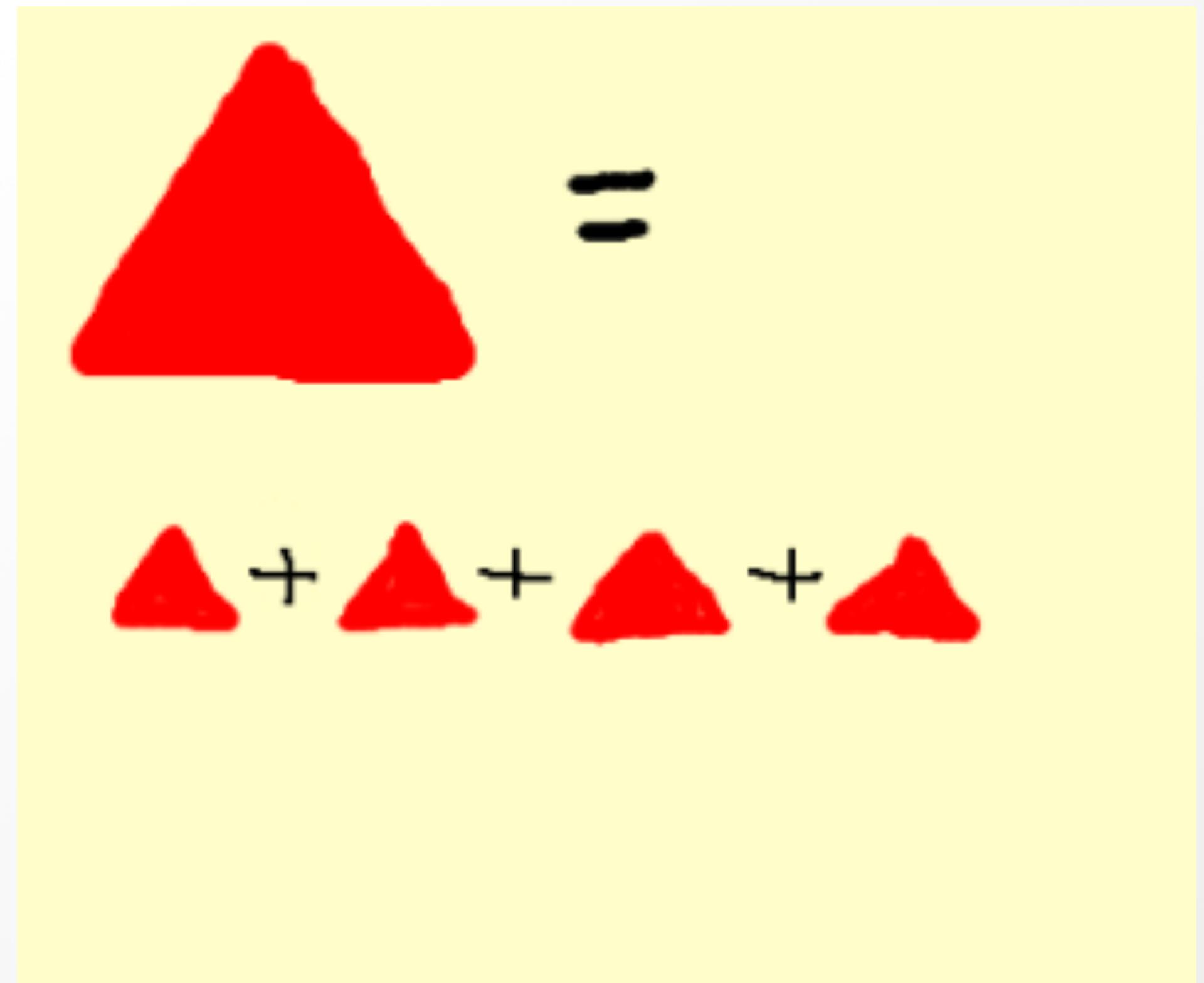
The Subsemble Algorithm

- An example of training time for a Super Learner and a Subsemble using the same three base learners.
- These were run on a 32-core machine, such that all the base (sub)models could be trained in parallel.
- These benchmarks were executed using the `subsemble` R package.



The Subsemble Algorithm

- Subsemble allows you to “squeeze” the machine learning task into pieces small enough to run on your hardware – just select the appropriate number of partitions (subsets).
- If you have heterogenous training data, you may choose to partition your data into subsets using a clustering algorithm.
- There is an extension of the Subsemble Algorithm called “Supervised Regression Tree (SRT) Subsemble Algorithm” which can learn the optimal subsets.



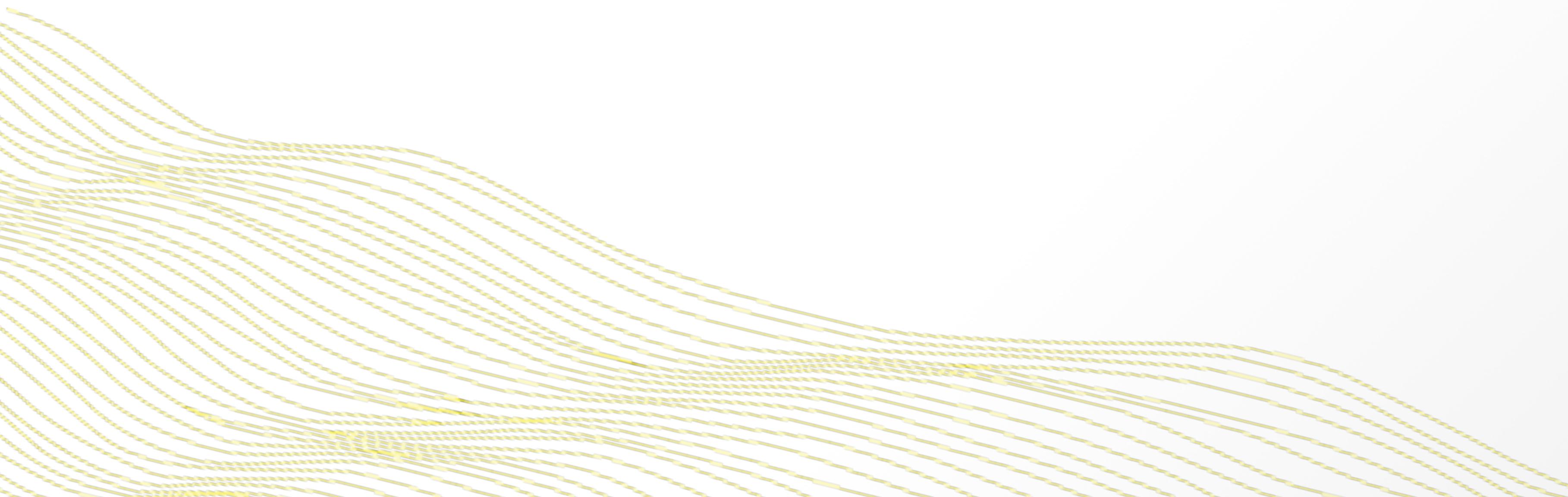
The Subsemble Algorithm

Stephanie Sapp, Mark J. van der Laan & John Canny.

Subsemble: An ensemble method for combining subset-specific algorithm fits. *Journal of Applied Statistics*, 41(6):1247-1259, 2014.

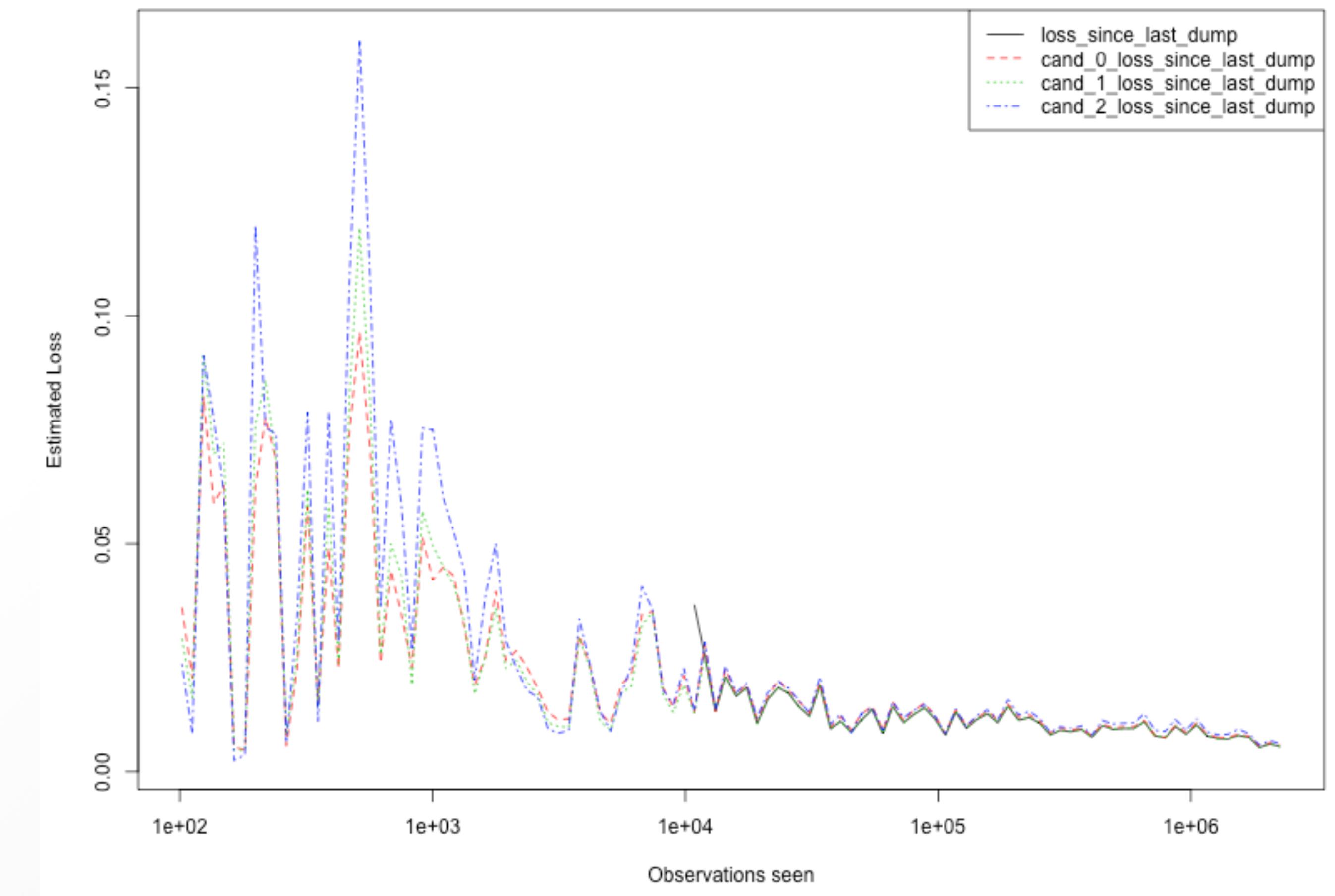
- Article: <http://dx.doi.org/10.1080/02664763.2013.864263>
- Preprint: <https://biostats.bepress.com/ucbbiostat/paper313>
- R package: <https://github.com/ledell/subsemble>

Online Super Learner



Online Super Learner

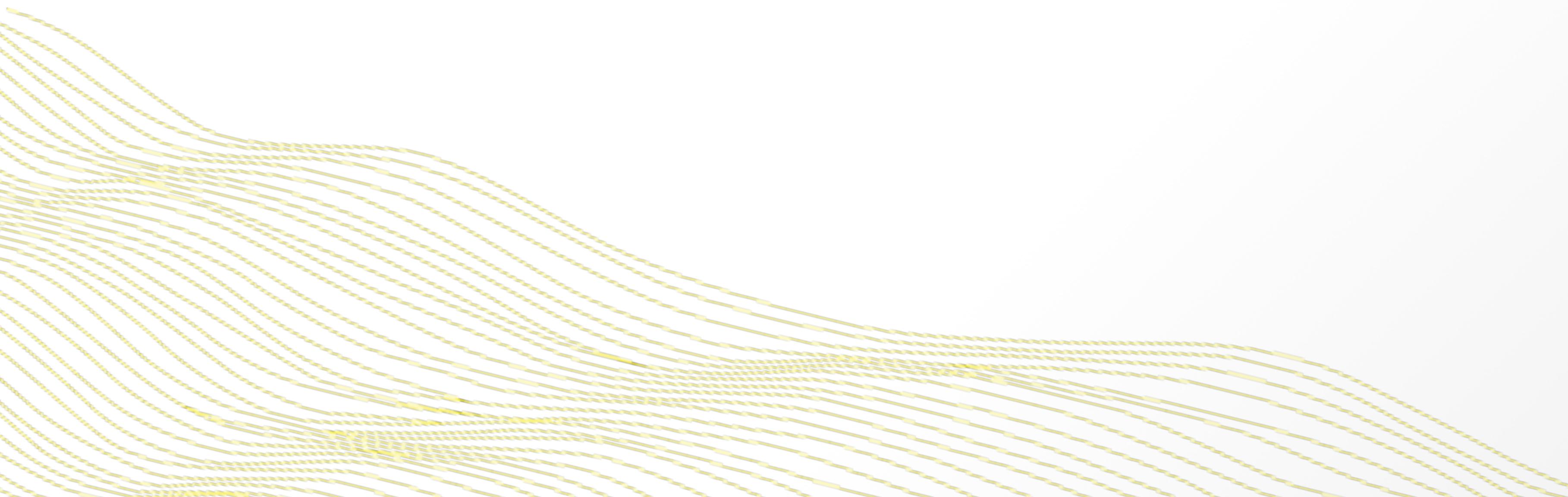
- The “Online Super Learner” is an experiment in extending stacking to an online setting.
- It was implemented on top of Vowpal Wabbit (VW) and uses VW base learners.
- For simplicity, the metalearner is a linear model, trained using the Non-Negative Least Squares (NNLS) algorithm.



Online Super Learner

- The primary distinction between Super Learner and Online Super Learner is in how the level-one data is generated.
- Single-pass mode: The predicted value for observation i is generated using the model trained on observations $1, \dots, i-1$. These predicted make up the level-one Z matrix.
- Multi-pass mode: Every v^{th} training example is put aside into a hold-out set (never trained on). Each training iteration produces the predicted values are generated for these samples and placed into the Z matrix.
- The most recent m rows of the Z matrix are used to update α , the candidate weight vector (in the case of a linear metalearner), which specifies the metalearning fit at a given iteration.

H₂O Ensemble



H2O Overview

- H2O Ensemble is a scalable implementation of the Super Learner algorithm for H2O.
- H2O is an open-source, distributed machine learning library written in Java with APIs in R, Python, Scala and REST/JSON.
- Produced by H2O.ai in Mountain View, CA.
- H2O.ai advisers are Trevor Hastie, Rob Tibshirani and Stephen Boyd from Stanford.



H2O Ensemble Overview



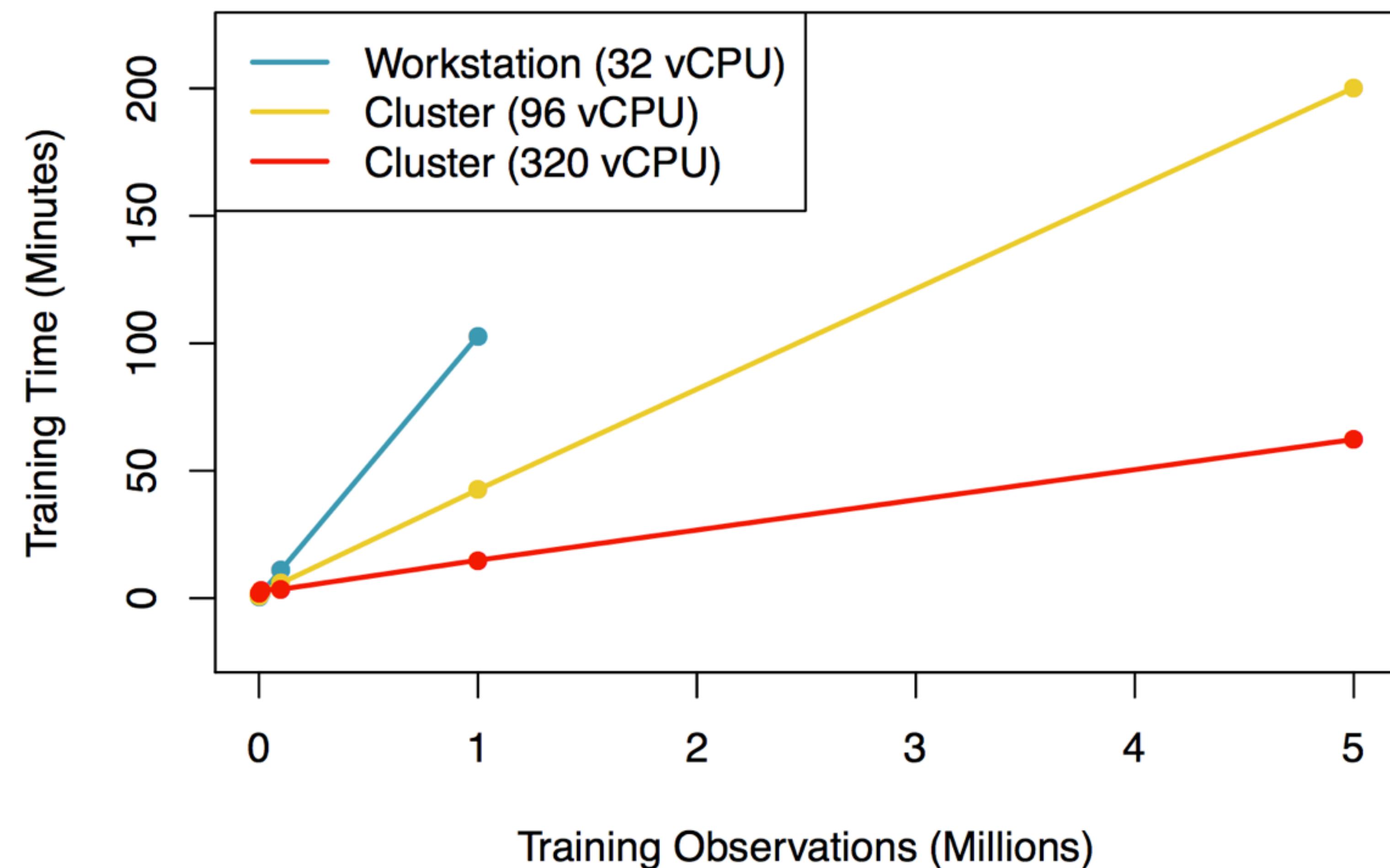
Using H2O as the base software provides incredible speed-up over R-based packages like SuperLearner and subsemble.

- H2O Ensemble is currently available as an R package, extending the h2o R API, and other APIs are in development.
- Implements regression and binary classification. Multi-class support is in development.
- Utilities to generate diverse sets of base learners for better performance (e.g. Random Grid Search).

H2O Ensemble Overview

- To scale, just add nodes to your H2O cluster.
- Also works just as easily on your laptop utilizing all available cores (same software runs single-node and multi-node clusters).

Runtime Performance of H2O Ensemble



H2O Ensemble Resources

H2O Ensemble training guide:

<http://tinyurl.com/learn-h2o-ensemble>

H2O Ensemble homepage on Github:

<http://tinyurl.com/github-h2o-ensemble>

H2O Ensemble R Demos:

<http://tinyurl.com/h2o-ensemble-demos>

Thank you!

@ledell on Github, Twitter
erin@h2o.ai

<http://www.stat.berkeley.edu/~ledell>