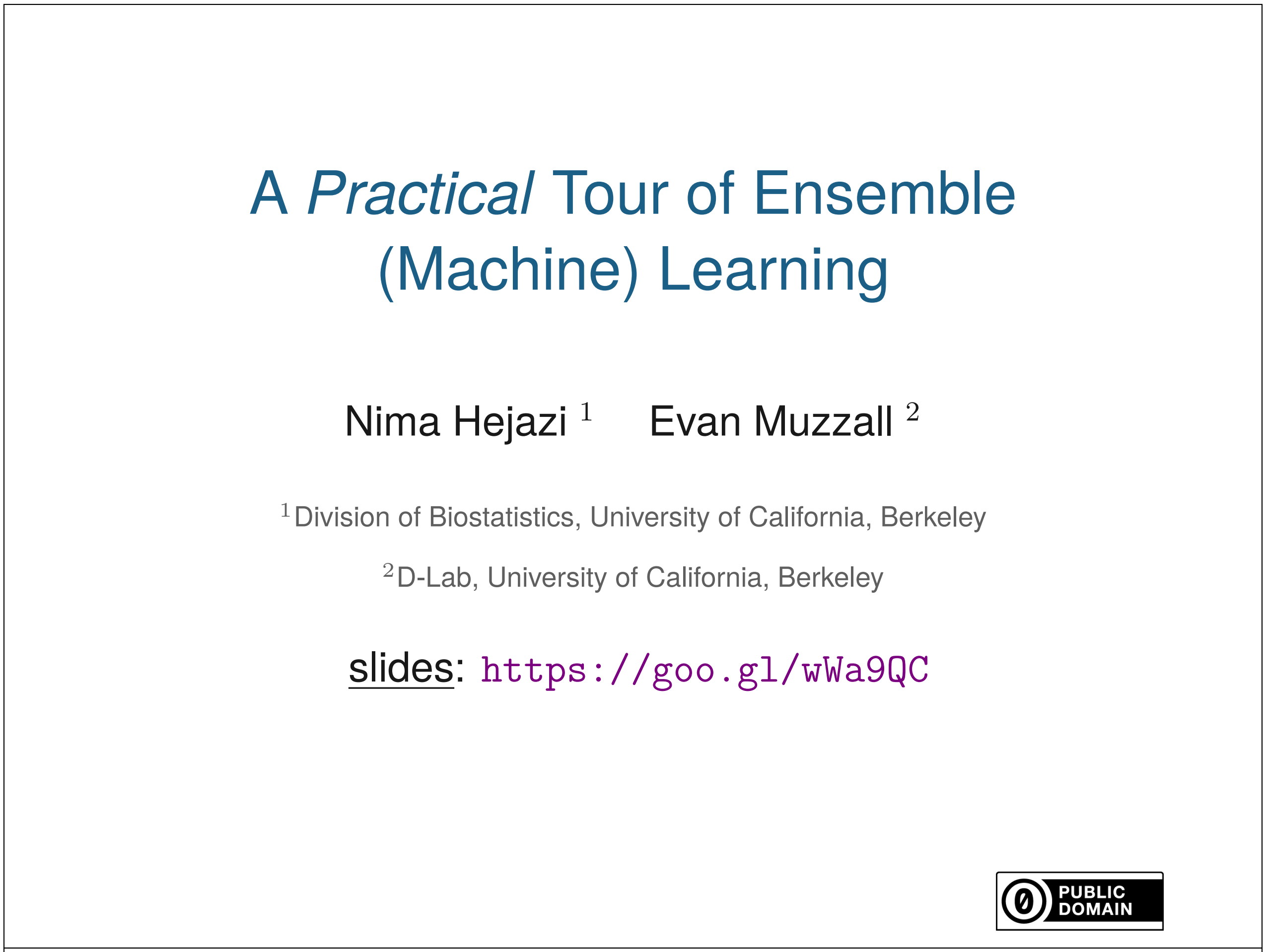


A *Practical* Tour of Ensemble (Machine) Learning

Nima Hejazi [1] Evan Muzzall [2]

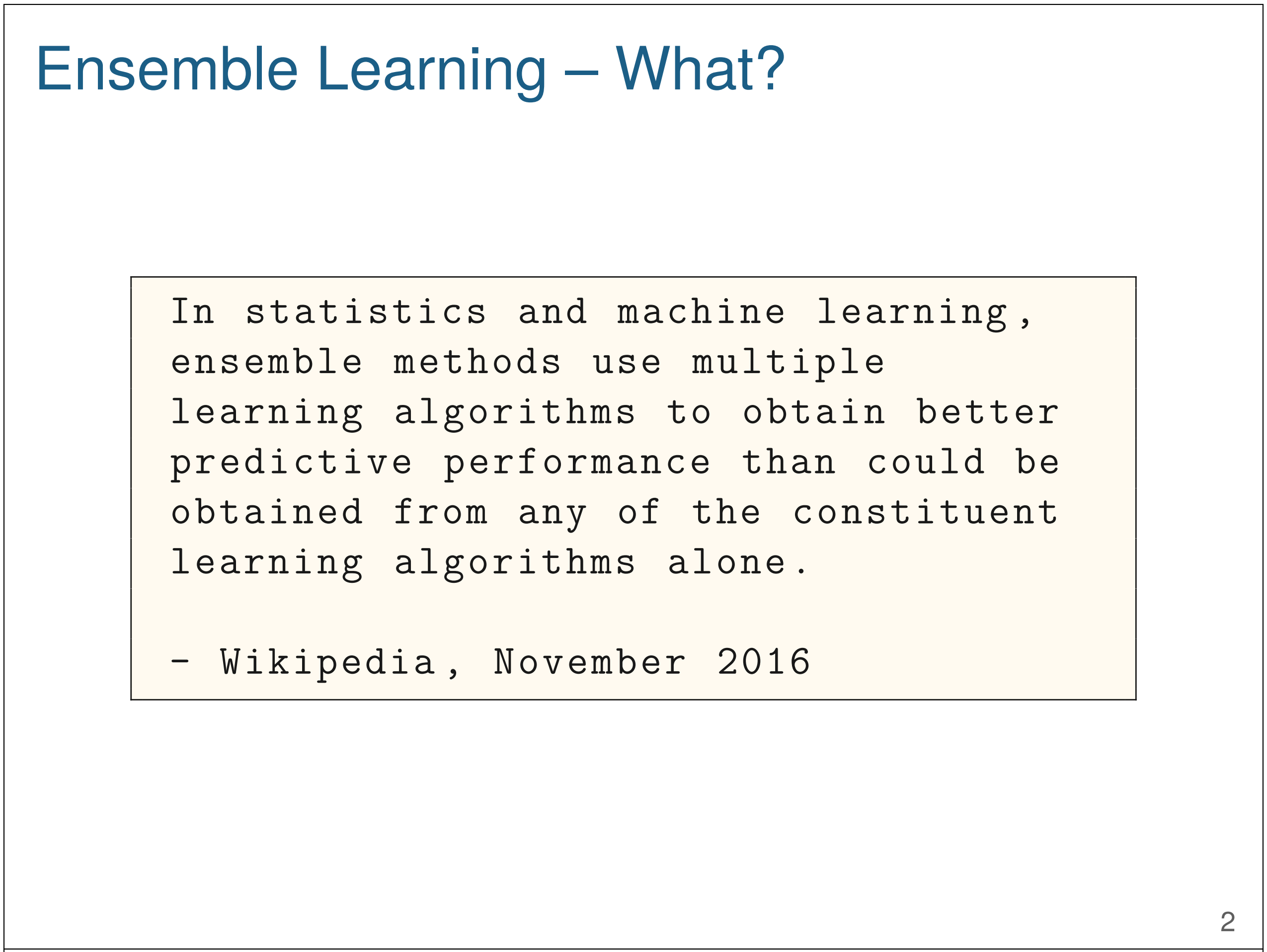
[1] Division of Biostatistics, University of California, Berkeley

[2] D-Lab, University of California, Berkeley

slides: https://goo.gl/wWa9QC

These are slides from a presentation on practical ensemble learning with the Super Learner and h2oEnsemble packages for the R language, most recently presented at a meeting of The Hacker Within, at the Berkeley Institute for Data Science at UC Berkeley, on 6 December 2016.

source: `https://github.com/nhejazi/talk-h2oSL-THW-2016`
slides: `https://goo.gl/CXC2FF`
with notes: `http://goo.gl/wWa9QC`



Ensemble Learning – What?

> In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.
>
> - Wikipedia, November 2016

This rather elementary definition of "ensemble learning" encapsulates quite well the core notions necessary to understand why we might be interested in optimizing such procedures. In particular, we will see that a weighted collection of individual learning algorithms can not only outperform other algorithms in practice but also has been shown to be theoretically optimal.

Ensemble Learning – Why?

- Ensemble methods outperform individual (base) learning algorithms.
- By combining a set of individual learning algorithms using a *metalearning* algorithm, ensemble methods can approximate complex functional relationships.
- When the true functional relationship is not in the set of base learning algorithms, ensemble methods approximate the true function well.
- *n.b.*, ensemble methods can, even asymptotically, perform only as well as the best weighted combination of the candidate learners.

A variety of techniques exist for ensemble learning, ranging from the classic “random forest” (of Leo Breiman) to “xgboost” to “Super Learner” (van der Laan et al.). In this talk, we will focus on the elementary theoretical properties of “Super Learner”, with an eye towards application.

Theoretically, a range of different algorithms can be used in the metalearning step; however, in practice, often, logistic regression is used.

Ensemble Learning – How?

Common strategies for performing ensemble learning:

- **Bagging** – reduces variance and increases accuracy; robust against outliers; often used with decision trees (*i.e.*, Random Forest).
- **Boosting** – reduces variance and increases accuracy; not robust against outliers or noise; accomodates any loss function.
- **Stacking** – used in combining "strong" learners; requires a *metalearning* algorithm to combine the set of learners.

While a number of different strategies exist for combining various types of learning algorithms, most modern methods rely on stacking to produce powerful ensemble learners. These sorts of ensemble learners are what you want to use to win Kaggle competitions!

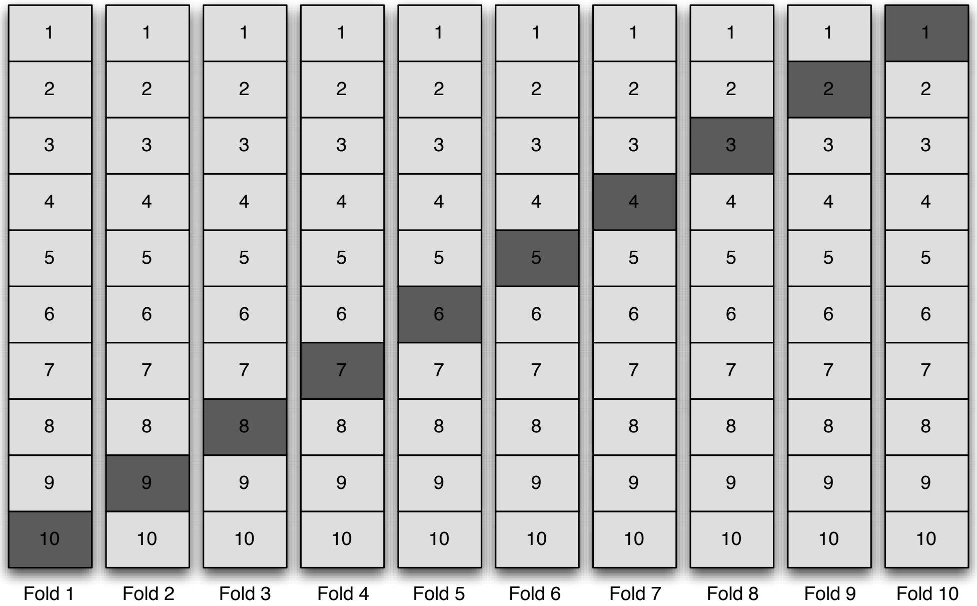
Introduction to Super Learner

- 1996 paper “Stacked Regressions” (L. Breiman) introduced the notion of model stacking using k-fold cross-validation, the precursor to the modern Super Learner algorithm.
- 2007 paper “Super Learner” (van der Laan *et al.*) worked out theoretical details on the optimality of stacking. Before this, the reasons for the superb performance of stacking were unknown.
- The Super Learner algorithm learns the optimal combination of the base learner fits in a manner that is provably asymptotic optimal.

The Super Learner algorithm allows researchers to use multiple algorithms to outperform a single algorithm in realistic non-parametric and semi-parametric statistical models that are based on actual knowledge.

The term algorithm is used very loosely to describe any mapping from the data into a predictor. This can range from a simple logistic regression to more complex algorithms such as neural nets.

Interlude: Cross-Validation



The validation set rotates V times such that each set is used as the validation set once.

Cross-validation solves the problem of having many algorithms, and not knowing which one to use and helps us avoid overfitting.

For any given fold, V—1 sets will comprise the training set and the remaining 1 set is the validation set.

The observations in the training set are used to construct (or train) the candidate estimators.

The observations in the validation set are used to assess the performance (i.e., risk) of the candidate algorithms.

Optimality of Super Learner

For a random variable $O = (W, A, Y)$, let the **oracle selector** be a rule that picks the algorithm with the lowest cross-validated risk under the *true probability distribution* P_0. The **oracle selector** is unknown because it depends on observed data and the truth.

Asymptotic results prove that in realistic scenarios (where none of the algorithms represent the true relationship), the "discrete super learner" performs *asymptotically as well as* the oracle selector (the best estimator given the algorithms in the collection).

To clarify, theory shows that that the discrete super learner performs as well as the oracle selector, up to a second order term. The loss function must be bounded, and then we will perform as well as the algorithm that is the risk minimizer of the expected loss function. The number of algorithms in the library can grow with sample size.

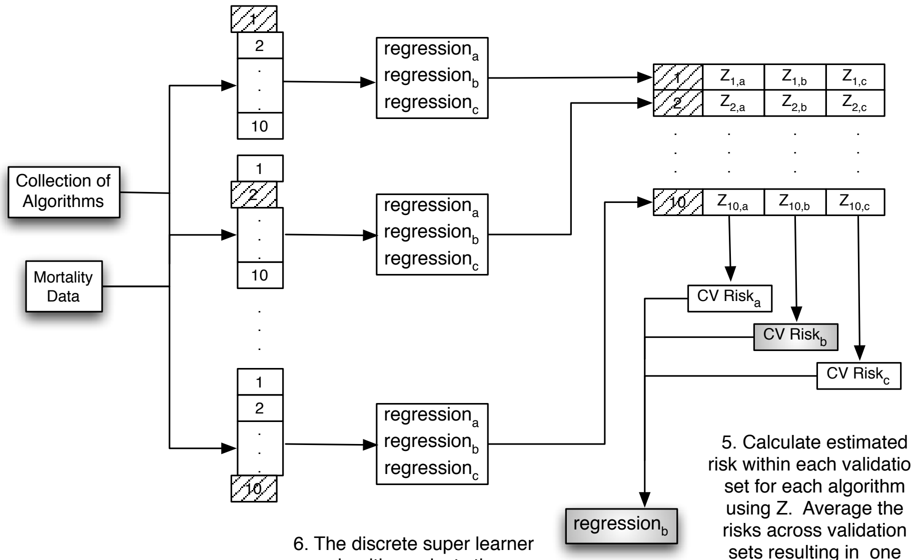
The Discrete Super Learner

1. Input data and a collection of algorithms.

2. Split data into 10 blocks.

3. Fit each of the 3 algorithms on the training set (non-shaded blocks).

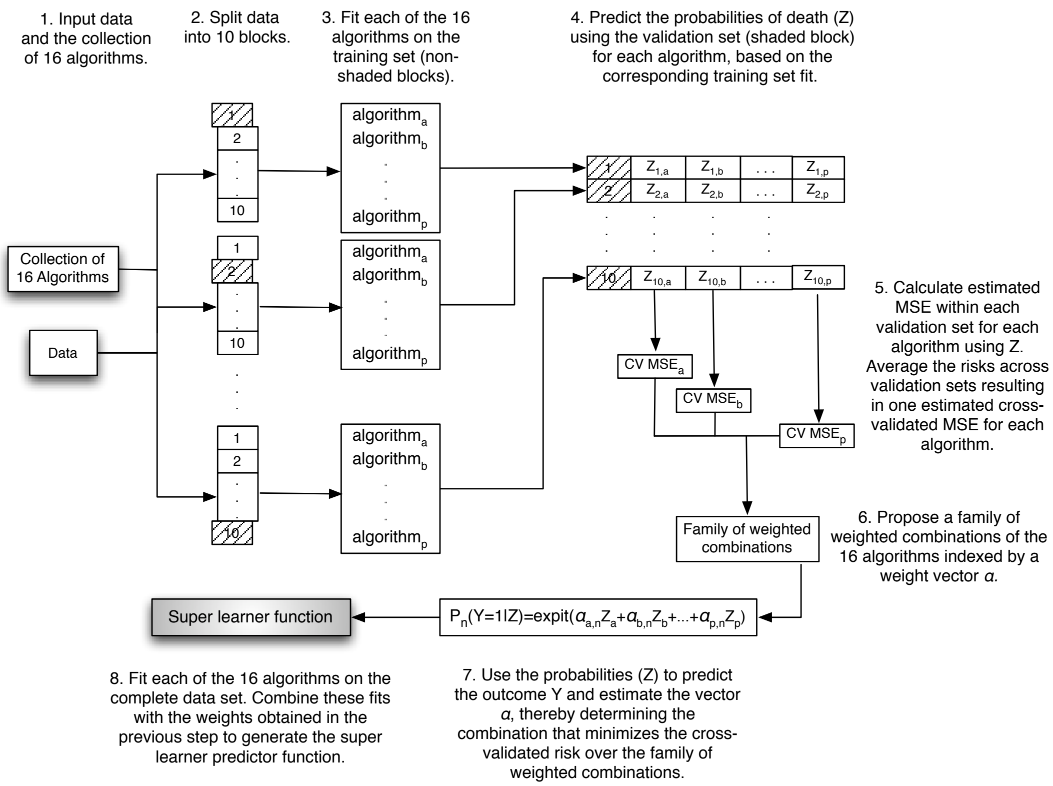
4. Predict the estimated probabilities of death (Z) using the validation set (shaded block) for each algorithm, based on the corresponding training set fit.



5. Calculate estimated risk within each validation set for each algorithm using Z. Average the risks across validation sets resulting in one estimated cross-validated risk for each algorithm.

6. The discrete super learner algorithm selects the algorithm with the smallest cross validated risk

The Super Learner Algorithm



1. Input data and the collection of 16 algorithms.

2. Split data into 10 blocks.

3. Fit each of the 16 algorithms on the training set (non-shaded blocks).

4. Predict the probabilities of death (Z) using the validation set (shaded block) for each algorithm, based on the corresponding training set fit.

5. Calculate estimated MSE within each validation set for each algorithm using Z. Average the risks across validation sets resulting in one estimated cross-validated MSE for each algorithm.

6. Propose a family of weighted combinations of the 16 algorithms indexed by a weight vector α.

7. Use the probabilities (Z) to predict the outcome Y and estimate the vector α, thereby determining the combination that minimizes the cross-validated risk over the family of weighted combinations.

8. Fit each of the 16 algorithms on the complete data set. Combine these fits with the weights obtained in the previous step to generate the super learner predictor function.

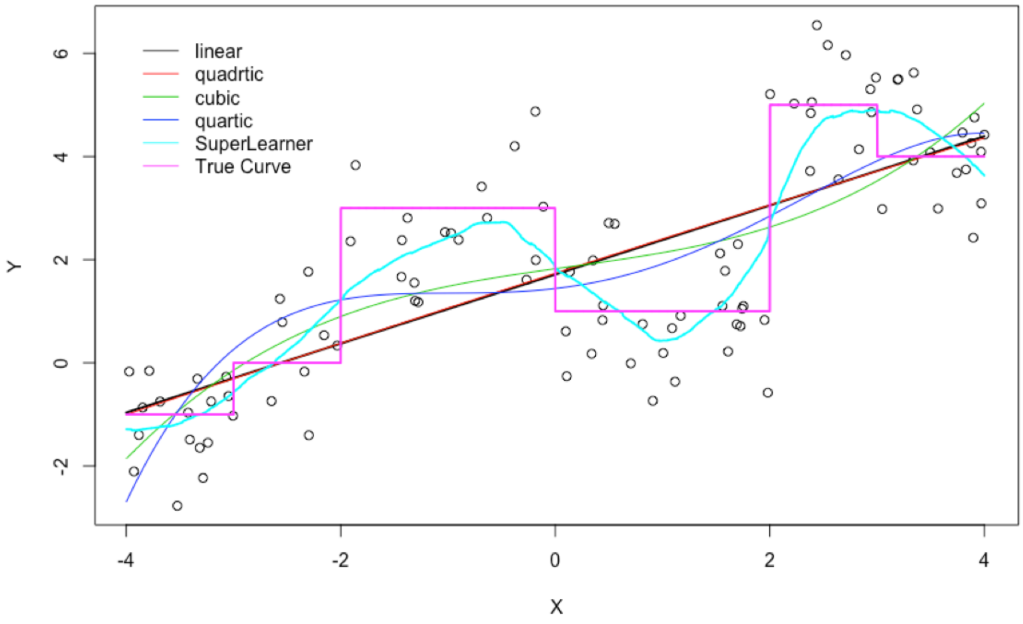
1. Cross-validate base learners:

- Perform k-fold cross-validation on each learner and collect the cross-validated predicted values from each of the L algorithms.
- The N cross-validated predicted values from each of the L algorithms can be combined to form a new $N \times L$ matrix. Call this matrix, with the original response vector, "level-one" data.

2. Metalearning:

- Train the metalearning algorithm on the "level-one" data.
- Train each of the L base algorithms on the full training set.
- The "ensemble model" consists of the L base learning models and the metalearning model, which can be used to generate predictions on a test set.

Ensembles with Super Learner



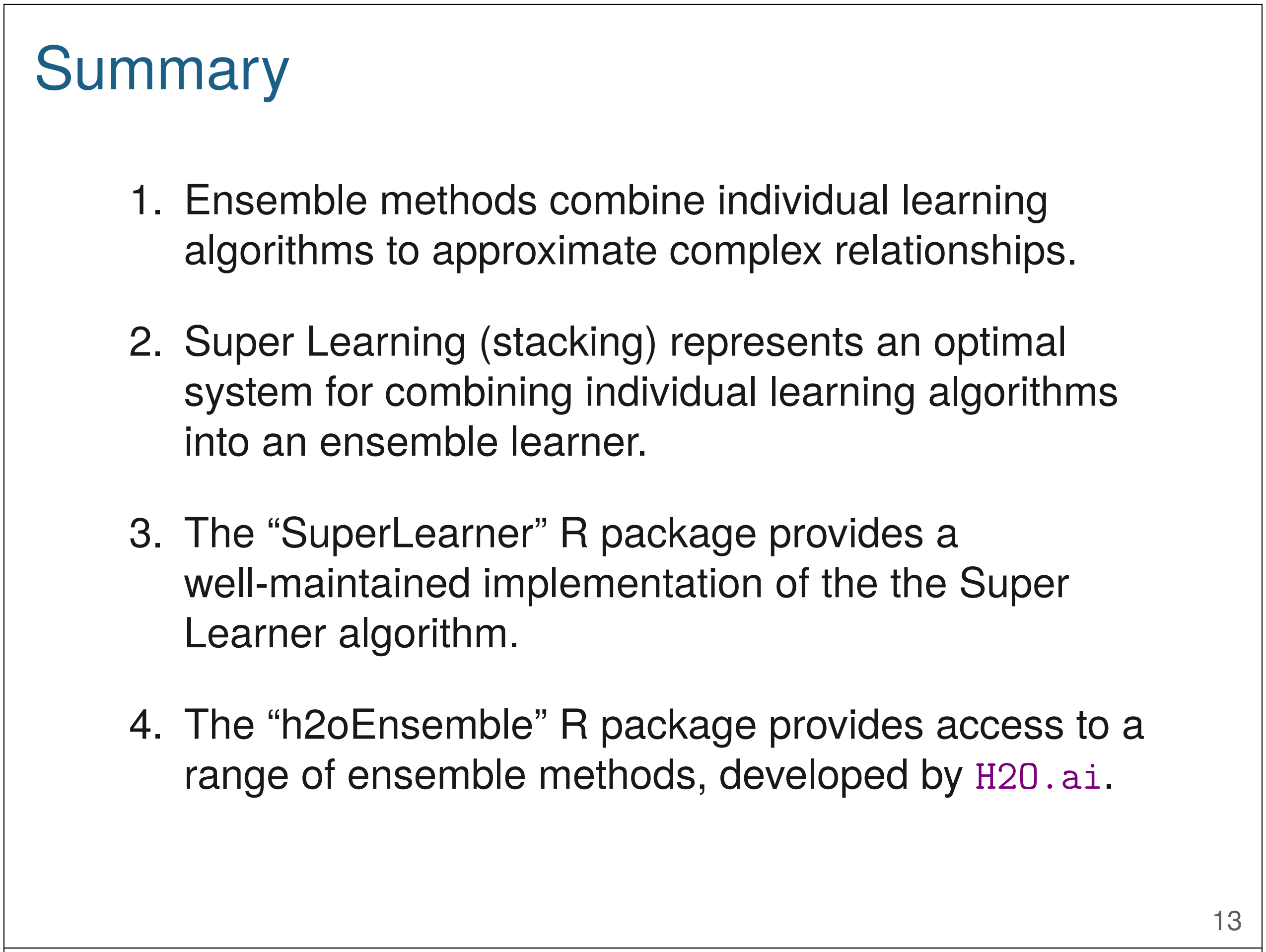
R Package: “SuperLearner”

- Implements the Super Learner prediction method (stacking) and contains a library of prediction algorithms to be used in the Super Learner.
- Provides a clean interface to numerous algorithms in R and defines a consistent API for extensibility.

R Package: “h2oEnsemble”

Extension to the “h2o” R package that allows the user to train an ensemble in the H2O cluster using any of the supervised machine learning algorithms in H2O.

- Uses data-distributed and parallelized Java-based algorithms for the ensemble.
- All training and data processing are performed in the high-performance H2O cluster.
- Supports regression and binary classification.



Summary

1. Ensemble methods combine individual learning algorithms to approximate complex relationships.
2. Super Learning (stacking) represents an optimal system for combining individual learning algorithms into an ensemble learner.
3. The “SuperLearner” R package provides a well-maintained implementation of the the Super Learner algorithm.
4. The “h2oEnsemble” R package provides access to a range of ensemble methods, developed by `H2O.ai`.

Just a summary of what we discussed today.

Slides: `http://goo.gl/wWa9QC`

GitHub: `nhejazi/talk-h2oSL-THW-2016`

Here’s where you can find the resources prepared for this talk.