Ensemble Methods for Classification

Ian Johnson

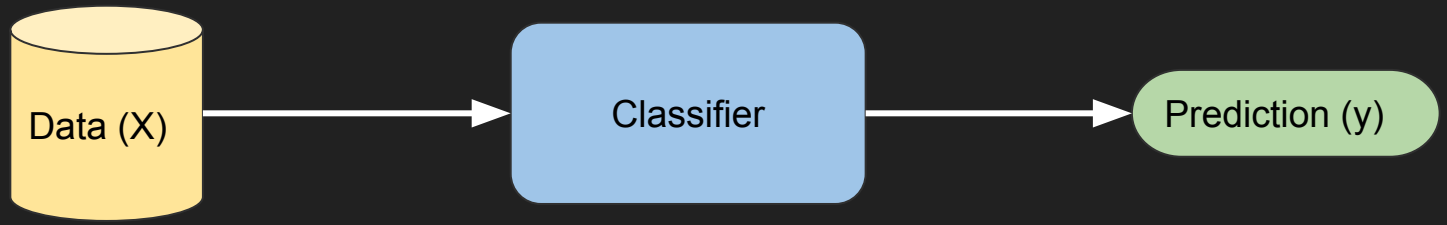
Schedule

- What is Ensemble Classification?
  - Associative Classification Example
- Bootstrap Aggregating (Bagging)
  - Random Forests Example
- Boosting
  - Adaboost Example
- Stacking
  - RCAR Example

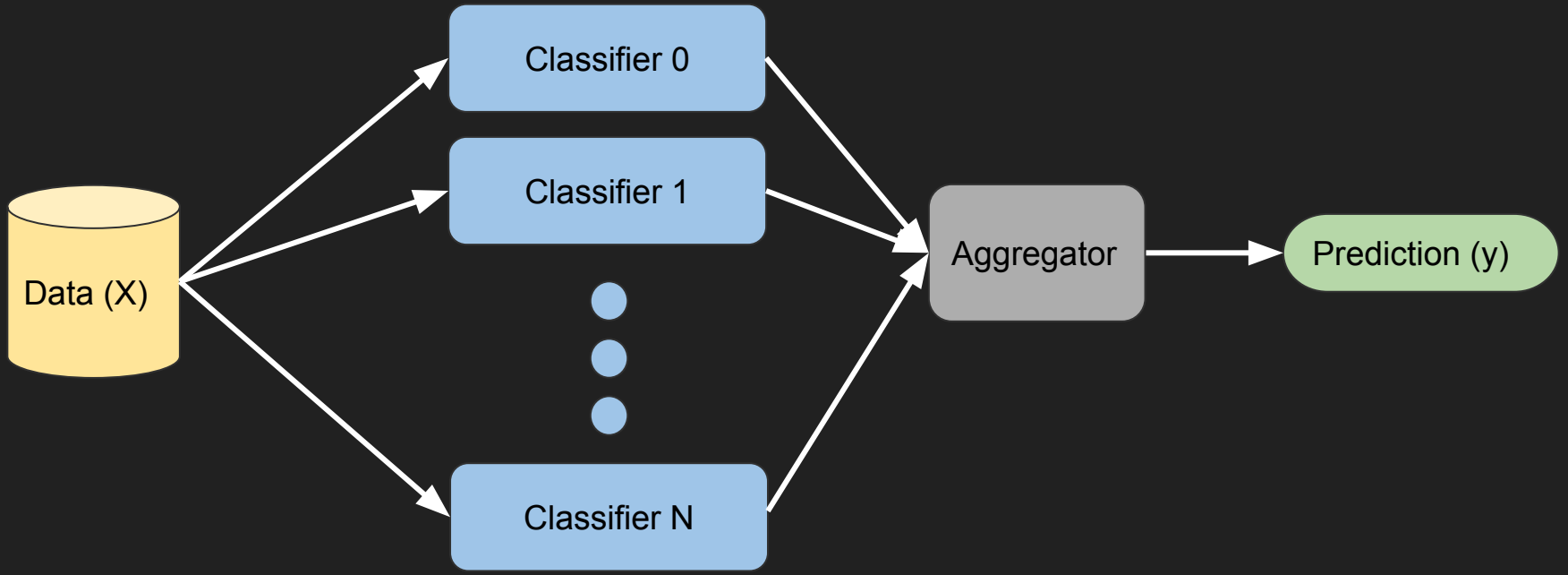
Schedule

- **What is Ensemble Classification?**
  - Associative Classification Example
- Bootstrap Aggregating (Bagging)
  - Random Forests Example
- Boosting
  - Adaboost Example
- Stacking
  - RCAR Example

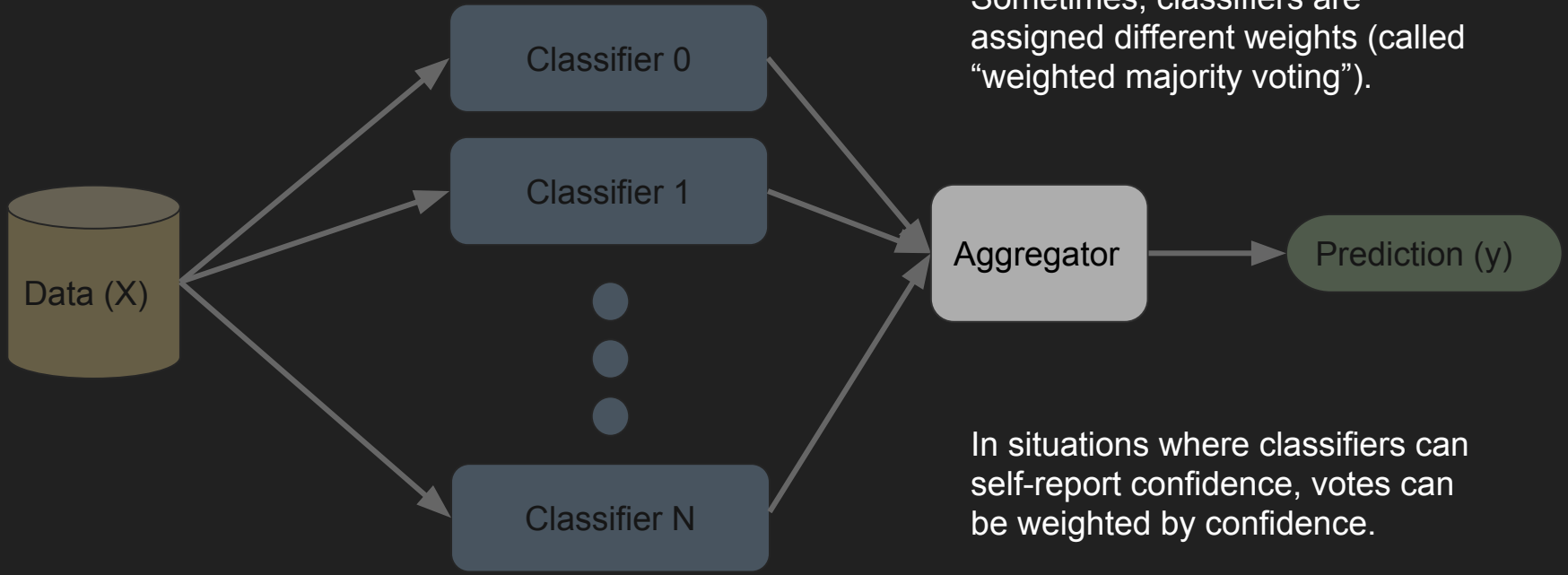
Traditional Classification



Ensemble Classification



Ensemble Classification

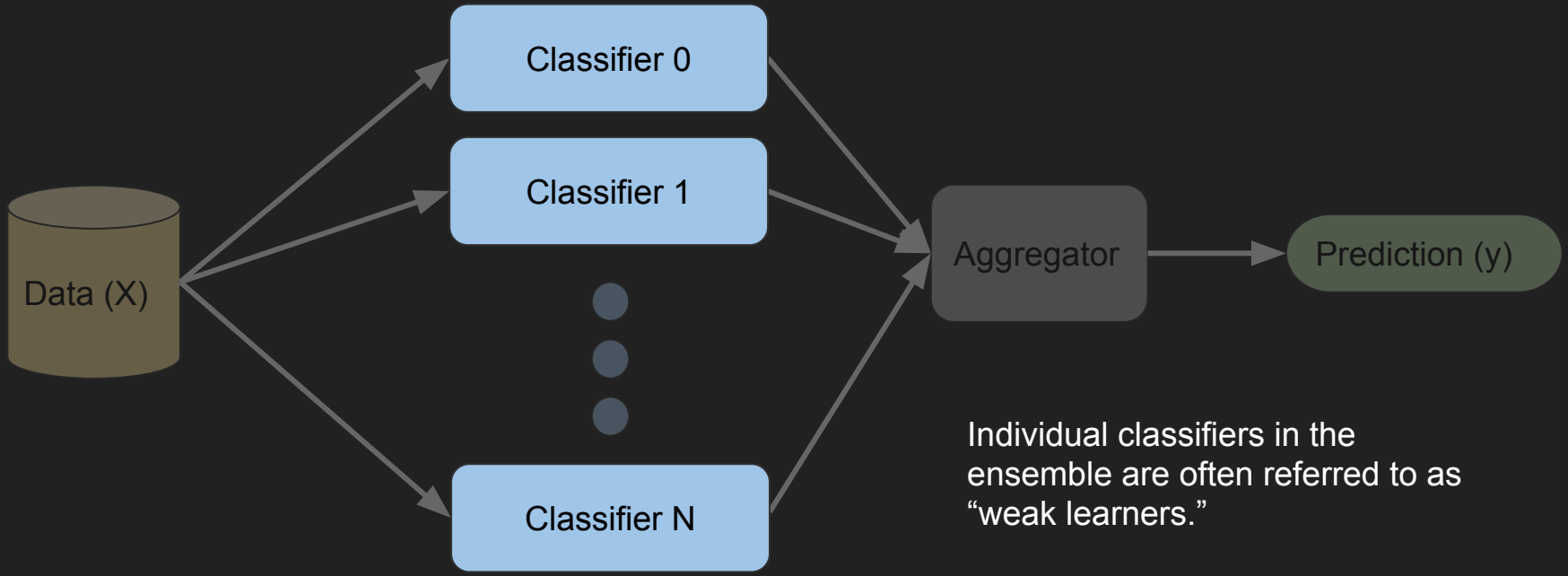


Aggregation is typically performed with some form of majority voting.

Sometimes, classifiers are assigned different weights (called “weighted majority voting”).

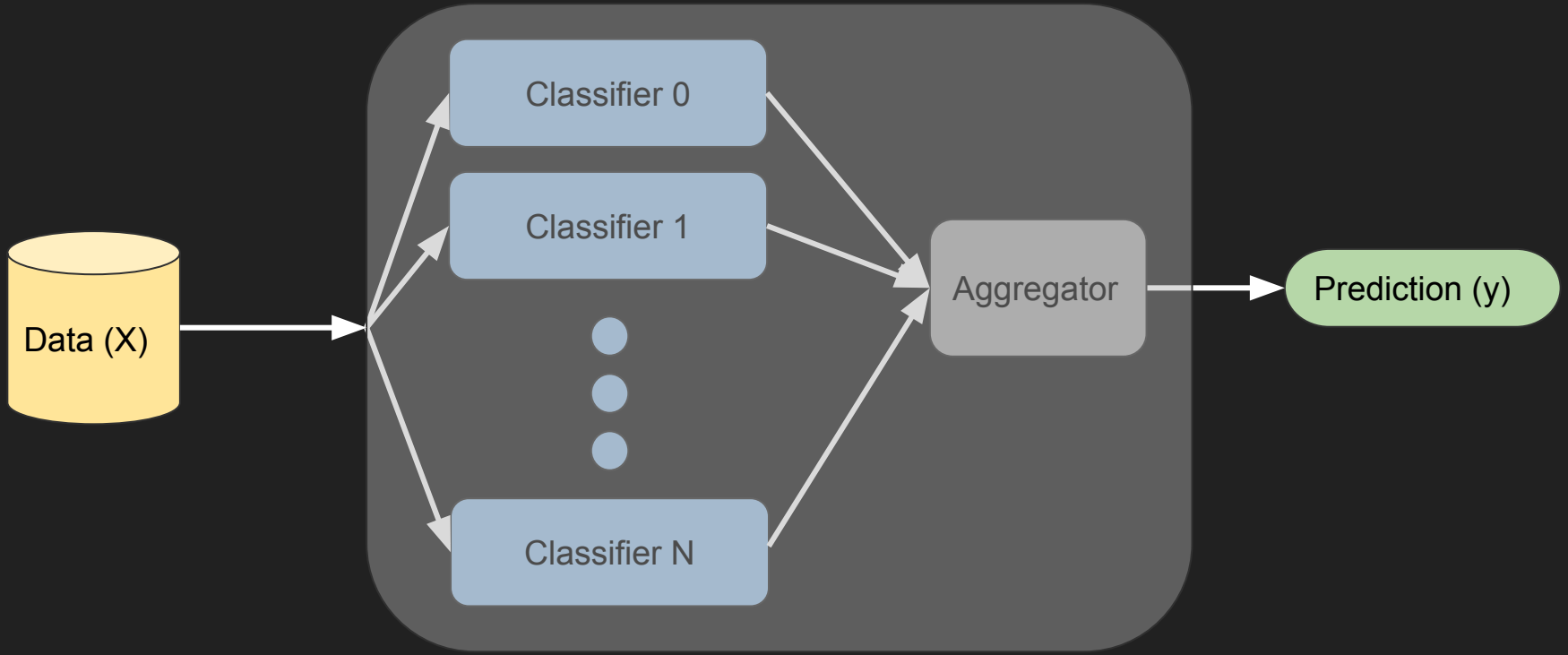
In situations where classifiers can self-report confidence, votes can be weighted by confidence.

Ensemble Classification



Individual classifiers in the ensemble are often referred to as “weak learners.”

Ensemble Classification



Why Should We Do This?

- Large sets of weak learners are more effective than 1 strong learner
- Less risk of overfitting
- Effective for anomaly detection
  - A single strong learner will learn to ignore anomalies, but a small set of weak learners will often learn to focus on them
- Often more computationally feasible
  - Cost of refining the hypothesis of any learner scales superlinearly with the current value of the hypothesis
- Provides some really simple mechanisms to handle class imbalance

Schedule

- What is Ensemble Classification?
  - **Associative Classification Example**
- Bootstrap Aggregating (Bagging)
  - Random Forests Example
- Boosting
  - Adaboost Example
- Stacking
  - RCAR Example

Association Rules

An association rule is an observed trend in transaction data where the presence some set of items is associated the presence of some other set of items.

	Chips	Milk	Salsa	Tortillas	Chicken	Beans	Beer
1	1		1				1
2		1	1	1		1	
3		1			1		1
4	1		1	1			1
5		1		1	1		

Association Rules

Chips + Salsa → Beer

Antecedent (LHS) → *Consequent (RHS)*

	Chips	Milk	Salsa	Tortillas	Chicken	Beans	Beer
1	1		1				1
2		1	1	1		1	
3		1			1		1
4	1		1	1			1
5	1	1		1	1		

Association Rules

Antecedents and Consequences can have any arbitrary number of items.

Support is the fraction of rows in the transaction data that contain the (antecedent, consequent) pair.

Confidence is the conditional probability of the consequent, given the antecedent.

Lift is a measure of the likelihood that the antecedent and consequent are statistically dependent on one another. A lift of 1 means that they are independent.
A lift of >1 means the antecedent increases the probability of the consequent.
A lift of <1 means the antecedent decreases the probability of the consequent.

Classification with Association Rules (CBA)

Class Association Rules (CARs) are a special type of association rule where the consequent is a single item that represents the class variable in a classification problem.

CBA was first introduced in a paper called *Integrating Classification and Association Rule Mining.*

We’re going to ignore the details of how this algorithm actually works, because we’re going to read that paper after Spring Break.

Code Demo

R Package arulesCBA

Schedule

- What is Ensemble Classification?
  - Associative Classification Example
- **Bootstrap Aggregating (Bagging)**
  - Random Forests Example
- Boosting
  - Adaboost Example
- Stacking
  - RCAR Example

Bootstrap Aggregating (Bagging)

Bagging is an equal-vote ensemble classification approach.

In bagging, an arbitrary number of weak learners is trained, each on a random sample of the data.

The random subsampling of the training data to train each learner promotes model variance.

This simplest, most canonical example of bagging is random forests

Schedule

- What is Ensemble Classification?
  - Associative Classification Example
- Bootstrap Aggregating (Bagging)
  - **Random Forests Example**
- Boosting
  - Adaboost Example
- Stacking
  - RCAR Example

Random Forests

Random Forests are a combination of bagging and decision trees for ensemble classification.

Decision trees are simple, rule-based classifiers. For the sake of time, we’re going to ignore how they work and just think of them as simple, not-so-smart classifiers.

Code Demo

R Package randomForest

Schedule

- What is Ensemble Classification?
  - Associative Classification Example
- Bootstrap Aggregating (Bagging)
  - Random Forests Example
- **Boosting**
  - Adaboost Example
- Stacking
  - RCAR Example

Boosting

Boosting is an ensemble classification method which focuses on error correction by emphasizing previously-mislabeled data when training new models.

The general concept of boosting is that if the current set of weak learners mislabels a given training row, we should bias new weak learners toward labelling it correctly, over other training rows.

Boosting is a very successful strategy, but, for obvious reasons, is prone to overfitting, as the technique will emphasize correctly making predictions for noisy training data.

Boosting is typically a weighted-majority vote approach to ensemble classification.

Schedule

- What is Ensemble Classification?
  - Associative Classification Example
- Bootstrap Aggregating (Bagging)
  - Random Forests Example
- Boosting
  - **Adaboost Example**
- Stacking
  - RCAR Example

Adaptive Boosting (AdaBoost)

AdaBoost is a very popular boosting approach where a random sample of training rows are used to teach every weak learner.

When a new weak learner is trained, a biased sampling strategy is used to select training rows. Rows which have been mislabeled by existing weak learners are more likely to be sampled, while rows that are correctly labeled by existing weak learners are less likely to be sampled.

AdaBoost is typically implemented with decision trees as their weak learners, but any classifier can be used.

Code Demo

R Package fastAdaboost

Schedule

- What is Ensemble Classification?
  - Associative Classification Example
- Bootstrap Aggregating (Bagging)
  - Random Forests Example
- Boosting
  - Adaboost Example
- **Stacking**
  - RCAR Example

Stacking

Stacking is an approach to weighted-majority vote classification where the voting weights of weak learners are computed with another machine learning algorithm.

The general principle of stacking is to assign weights to weak learners to optimize the weighted-voting performance of the ensemble.

This can often be approximated with logistic regression, but can also be done with more complex techniques (such as with gradient descent).

Schedule

- What is Ensemble Classification?
  - Associative Classification Example
- Bootstrap Aggregating (Bagging)
  - Random Forests Example
- Boosting
  - Adaboost Example
- Stacking
  - **RCAR Example**

RCAR

RCAR is a new approach to associative classification (remember CBA?)

RCAR utilizes stacking to perform rule pruning and weighting after mining association rules

RCAR, like CBA, is from paper that we will read later this semester, so we’re going to skip the details.

As a high-level abstraction, think of RCAR as a stacking of association rule classification and logistic regression with lasso regularization to get rid of low-quality rules

Code Demo

R Package arulesCBA

Thank You for Your Attention.

References

Hahsler, Michael, et al. "The arules R-package ecosystem: analyzing interesting patterns from large transaction data sets." *Journal of Machine Learning Research* 12.Jun (2011): 2021-2025.

Rätsch, Gunnar, Takashi Onoda, and K-R. Müller. "Soft margins for AdaBoost." *Machine learning* 42.3 (2001): 287-320.

Wu, Dekai, Grace Ngai, and Marine Carpuat. "A stacked, voted, stacked model for named entity recognition." *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003.

Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.

Azmi, Mohamed, George C. Runger, and Abdelaziz Berrado. "Interpretable Regularized Class Association Rules Algorithm for Classification in a Categorical Data Space." *Information Sciences* (2019).

https://CRAN.R-project.org/package=fastAdaboost

https://CRAN.R-project.org/package=arulesCBA

https://CRAN.R-project.org/package=randomForest

Questions?