Ensembling: Learning Goals

1. Model Aggregation, Stacking
2. Boosting
3. Random Forests

- Russell 18.10, Tibshirani 8.8, 15

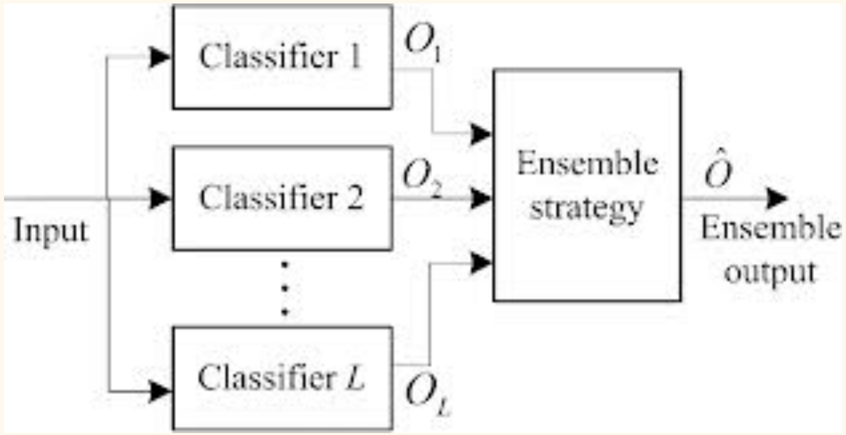
Many Classification Approaches

- Decision trees
- Linear models, e.g., regression, ridge, lasso
- Neural Nets
- Naïve Bayes
- kNN
- SVM
- Ensembles
- Random forests
- Hidden Markov Models
- Conditional Random Fields

Decision by "Committee"

- Prediction question: Are the Canucks going to the playoffs? Why?
- Many "experts", with their classification decision perhaps based on different perspectives
- More serious, e.g.
  - Is the Canadian economy growing? (classification)
  - How much will the Canadian economy grow in 2015? (regression)

Ensembling



- Given a training data set, we can build many classifiers
  - Using different parameters for a method (e.g., kNN), and/or using different methods
- On a test sample, all classifiers may not agree
- Which *one* to use? How about using them *all*?
- Aggregating from multiple classifiers is called ensembling

Combination/Aggregation Rules

- (classification) *m* classifiers, *m* binary votes
  - majority voting to create the ensemble decision
  - m typically an odd number
- (classification) *m* classifiers, *m* probabilities (positive if p > 0.5)
  - Averaging the *m* probabilities
  - E.g., averaging 0.45, 0.45, 0.9 gives 0.6 (1 strong positive trumps two weak negatives)
- (regression) average the *m* predicted values
- What about max/min?

When does ensembling help?

- If there is a strong classifier within the ensemble, adding weak ones into the ensemble may not help (and may backfire)
- If there are multiple weak classifiers, all somewhat independent, then aggregation helps
- But always a good idea to remove really weak classifiers (almost random guesses)

Model Selection for Ensembling

- Use CV to rank models
- In each iteration, remove the lowest performing ones (e.g., 1, or a fraction)
- Recurse until the performance of the ensemble stops improving
- But do we need to give *equal weights* to each model in an ensemble?

Stacking

- Based on the performance of individual models in CV, assign a weight to each model with the weights sum to 1
  - May also “regularize” by avoiding assigning too high weights to complex models
- In regression, one common way is to use the residuals in CV
- Special case: model selection in previous slide
  - non-selected models are given weight 0, with equal weights to all the selected models

Bagging (Bootstrapping)

- As we talked about "unequal" weights to models, why do we believe that *every training sample* be given an "equal" weight?
- And using every sample to train one classifier/model may lead to overfitting
- From the (large) training set, generate *m smaller* training sets, called bootstrap samples, by sampling with *replacement,* with each deriving a classifier
- Use ensembling on the *m* classifiers

Bagging (2)

- E.g., Bagging on 1NN classifiers
  - How is this different from mNN classifiers?
- How is CV different from bagging?
  - CV is used to *estimate performance* – not interested in reusing the classifiers for actual classification
  - If we were to keep the classifiers from the folds, there are still the difference of sampling with replacement vs partitioning the training dataset

Boosting

- In exactly the same way as we assign weights to different *models* in an ensemble, we can assign *weights* to different *examples* in the training set
  - Special case: bagging, samples in-or-out
- AdaBoost (Adaptive)
  - Assign the same weight to each example
  - But in CV, if an example is inaccurately predicted by the current model, increase its weight and redo
  - From m iterations, m classifiers/models derived
  - Ensembling the m classifiers

Boosting (2)

- Idea: to progressively make a classifier pay more attention to incorrectly classified examples – making the classifiers stronger and stronger (Russell Fig 18.33)
- Cons: may lead to overfitting, particularly if the training dataset contains noisy examples

Clicker Question

- Which is the most true?

a) Stacking cannot be used together with boosting.
b) Bagging cannot be used together with boosting.
c) When bagging and stacking are used together, boosting cannot be used.
d) Bagging, boosting and stacking can be used in any combination.

Online Learning

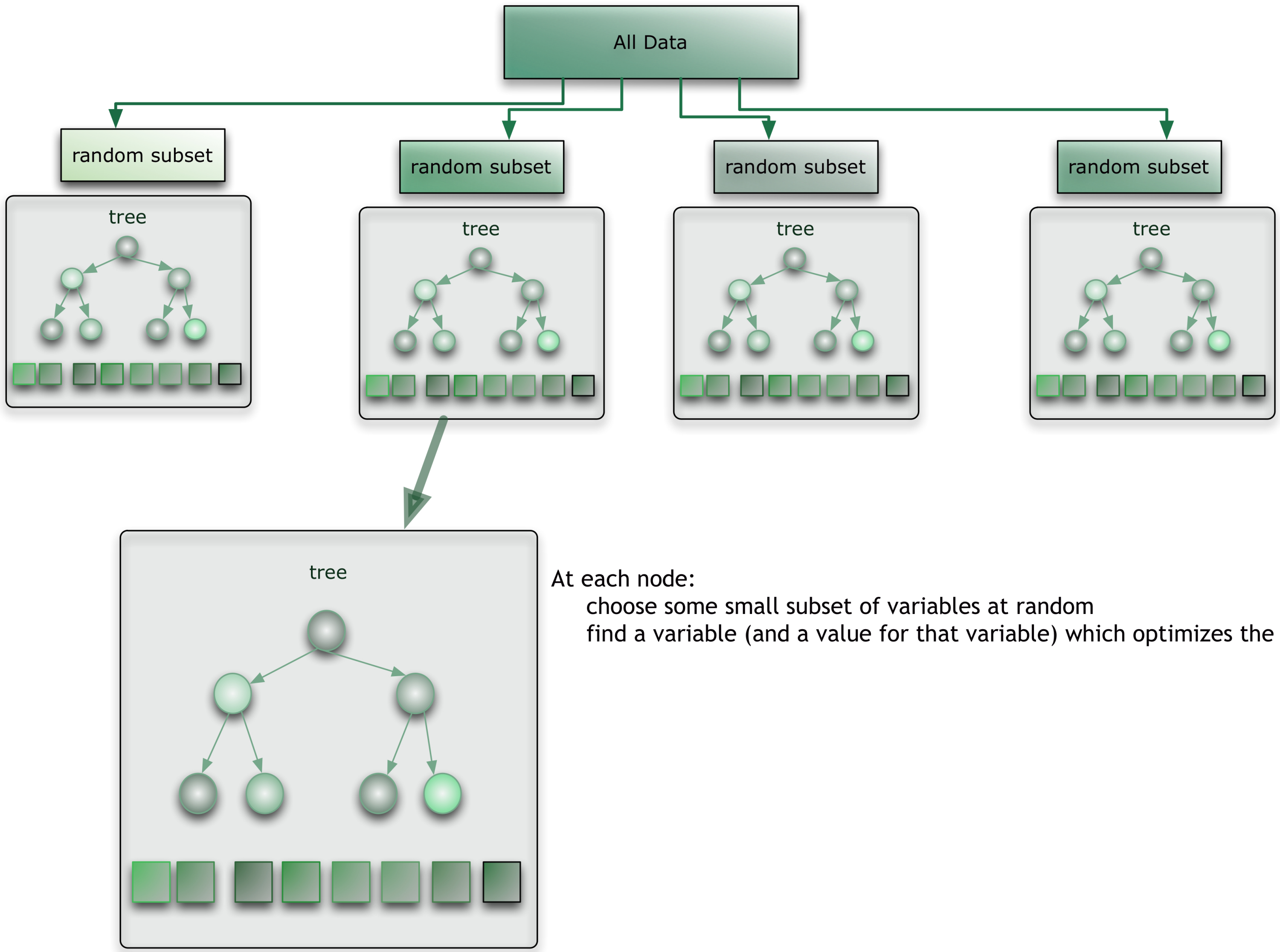
- Boosting is a special case of weighted training set learning
- So far we assume that the training dataset is static with one exception. (Which one?)
- What if we are learning something that is changing over time?
  - May want to assign lower weights to older examples
  - May give higher weights to models that are more accurate on the new examples

Random Forests

- Remember decision trees (e.g., split points)
- Remember bagging as a way to prevent overfitting by taking a subset of training examples
- Random forests build *m* trees by bagging
- In addition, to determine the winning split points, only a subset of features are considered
  - *Randomized in each node*
  - Called "feature bagging"
- Then ensembling the *m* trees (i.e., a *forest*)

Random Forests

-



At each node:

choose some small subset of variables at random

find a variable (and a value for that variable) which optimizes the split

Random Forests (2)

- Why feature bagging?
- Features can be "masked"by others
- Data may ensapsulate multiple mechanisms
- Feature bagging allows more features a better chance to exert their influence

Random Forests – Feature Importance

- Produces a score on the importance of each feature
- Use out-of-bag training examples that are not in the bootstrap sample
- The values of a feature in the bootstrap sample are permuted randomly
- Measure how much the out-of-bag error worsens across all the trees
- A feature deemed more important if the average out-of-bag error is higher

Concluding Remarks: Big Data

- Ensembling is widely used in many real-life complex applications to leverage from multiple classifiers
- For large training datasets, bagging is attractive
- Bagging, boosting and stacking can be combined in ways suitable for the applications
- Random forests is a popular ready-made ensemble