Reflection Week 4

Summary

This week's lectures explore methods that are used in classification tree models to make them more powerful: ensemble methods with random forest algorithm and AdaBoost. We also go further into the concept of assessing accuracy of these models by reintroducing some metrics and visualizations.

Concepts

- Random Forest (Bagging): averaging collections of very deep decision trees to minimize variance.
  - Works very well in a lot of cases, not just classification but also regression and clustering
  - Tends to require less hyper-parameter tuning
  - Trees can be learned in parallel
- Bootstrapping is the technique of randomly sampling with replacement to create datasets with the same size as the original dataset.
- Weak learner: a model that only slighly does better than random guessing
- AdaBoost: a model that use decision stumps (decision trees with only one level)

  Advantages:
  - powerful for real world datasets
  - higher maintenance (require hyper-parameter tuning)
  - expensive: sequential loop for each instance in the dataset, and take long time with big ones.
- Training AdaBoost is to train each model in succession:

  Star with the same weight for all points in the dataset: $\alpha_i = 1/N$

  For t in [1,2,..., T]:
  - Learn $\hat{f}_t(x)$ based on weights α_i for each instance in the dataset
  - Compute model weight $\hat{w}_t$: an accurate model should have a high weight (in absolute value)

    $$\hat{w}_t = \frac{1}{2}\ln\left(\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}\right)$$

    in which weight error is calculated as the sum of all weights of FP + FN cases
  - Recompute weights α_i : for all the things the model (decision stump) gets wrong, in crease the weight of that example, and decrease its weight otherwise.

    $$\alpha_i = \alpha_i e^{-\hat{w}_t} \text{ if } f_t(x_i) = y_i$$
    $$\alpha_i = \alpha_i e^{\hat{w}_t} \text{ if } f_t(x_i) \neq y_i$$
  - Normalize α_i by:

    $$\alpha_i \leftarrow \frac{\alpha_i}{\sum_{j=1}^{n} \alpha_j}$$
- AdaBoost theorem: training error of boosted classifier always reduces to 0 as T goes to infinity

- ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots True Positive Rate on False Positive Rate
- Precision-recall curve graphs precision against recall rates.
- Optimistic model will predict almost everything as positive: high recall, low precision
- Pessimistic model will predict almost everything as negative: low recall, high precision

Concerns

- How to choose between different ensemble methods?
- What do we mean when you say we "want to have high weight for models that are very accurate"? What is the criteria to be considered "high" vs. "low"? What models are there in this comparison, since decision stumps are sequentially added and not in parallel?