7 Problems and Solutions (Part 3 of 3)

Ensemble Learning

As simple models may perform poorly, in *ensemble learning* we combine many _____ models (each only a little better than random guessing) to get a strong *meta-model*.

Ensemble models work because good models agree on the same prediction, while (uncorrelated) bad models _____ on different ones.

_____ makes several models independently and averages their predictions to reduce overfitting:

- *Bagging* (short for "bootstrap aggregating") generates B random sets of examples $\{S_i : i = 1, \ldots, B\}$ of size N by _____ from the training data. For each S_i, train a decision tree f_i. Make a prediction:
  - for classification as the most frequent of the B predictions
  - for regression as the average of the B predictions, $\hat{y} = \hat{f}(\mathbf{x}) = \frac{1}{B}\sum_{i=1}^{B} f_i(\mathbf{x})$

- A *random forest* is a bagging variant that strives for uncorrelated trees by selecting a random subset of _____ for each tree. Hyperparameters include
  - the maximum _____ d of each tree
  - the _____ B of trees
  - the number of _____ to include per subset

  This is widely used and effective because it reduces overfitting of the ensemble model by under-emphasizing noise, outliers, and over- or under-represented examples in the data.

Boosting builds models _____ to reduce underfitting:

- *Boosting* iteratively creates models such that model $(i + 1)$ is trained to correct model i's errors by _____ training examples to increase the weight of mis-classified examples and decrease the weight of correctly-classified examples. The final *ensemble model* combines all the models.

- For *gradient boosting*:

  - For regression:

    * Start with a constant model $f = f_0(\mathbf{x}) = \dfrac{1}{N}\sum_{i=1}^{N} y_i$.

    * Calculate _____ $e_i = y_i - f(\mathbf{x}_i)$ for $i = 1, \ldots, N$.[1] Then train a new model f_1 with the original y values replaced by the _____ . The boosted model is then $f = f_0 + \alpha f_1$, where hyperparameter α is the *learning rate*.

    * Repeat by training f_2 on residuals with respect to f_1 and get the boosted model $f = f_0 + \alpha f_1 + \alpha f_2$, and so on, until making model f_M, where M is the maximum number of trees.

    Model $i + 1$ is trained to _____ of model i. Recall that in gradient descent we move our parameter vector by step size α opposite the direction of the gradient toward the value that minimizes an objective function. Gradient boosting uses _____ as a proxy for the gradient, as they show how to adjust the model to _____ . Again α limits the amount the model moves in one step. Burkov asserts the overall model f minimizes MSE.

    Hyperparameters include:

    * the number _____ of trees

    * the learning rate _____

    * the _____ of trees

    Boosting reduces _____ (where bagging reduced overfitting); _____ the depth and number of trees can help boosting avoid overfitting.

  - There is also gradient boosting for classification and for other tasks.

Gradient boosting is one of the best ML algorithms, as it gives accurate models and can handle _____ data sets. It usually outperforms random forests but is slower to train.

[1] Burkov calls this residual "$\hat{y}$," which I think must be a typo since we usually call a prediction $\hat{y}$.

Python

To learn more:

- User guide: https://scikit-learn.org/stable/modules/ensemble.html
- Reference manual:
  - bagging:
    - regression:
      https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html
    - classification:
      https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html
  - random forest:
    - regression:
      https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
    - classification:
      https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
    - plot decision surface:
      https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_iris.html
  - gradient boosting:
    - classification:
      https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
    - regression:
      https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html