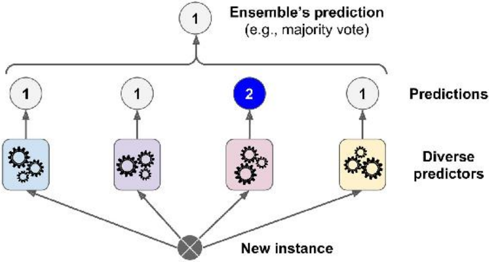
Ensemble Learning

Elliott Ash

Text Data Course, Bocconi 2018

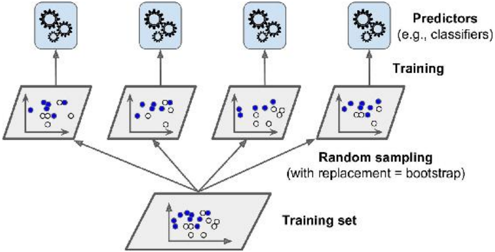
Voting Classifier



- voting classifiers generally out-perform the best classifier in the ensemble.
  - a "condorcet jury theorem" for machine learning
  - more diverse algorithms will make different types of errors, and improve your ensemble's robustness.

Bagging and Pasting

- Rather than use the same data on different classifiers, one can use different subsets of the data on the same classifier:

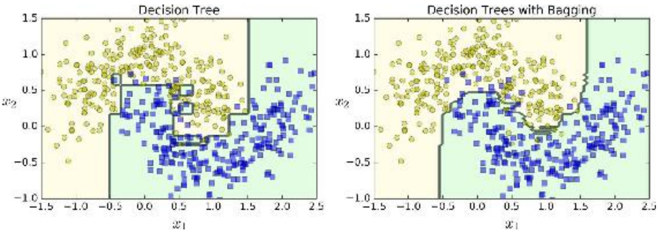


  - This is called **bagging** (bootsrap aggregating, when sampling with replacement) or **pasting** (when sampling without replacement).
- The ensemble predicts by aggregating the predictions:
  - for classification, use the most frequent prediction
  - for regression, use the average output

Bagging Benefits

- While the individual predictors have a higher bias than a predictor trained on all the data, aggregation reduces both bias and variance.
  - Generally, the ensemble has a similar bias but lower variance than a single predictor trained on all the data.
- Predictors can be trained in parallel using separate CPU cores.

Bagging in sklearn




```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier

bag_clf = BaggingClassifier(
        DecisionTreeClassifier(), n_estimators=50,
        max_samples=100, bootstrap=True, n_jobs=-1
    )

cross_val_score(bag_clf, X, Y).mean()
```

Out-of-bag Evaluation

- The `BaggingClassifier` samples a subset of data, and the remanining training instances (out-of-bag (oob) instances) can be used as a validation set.
  - In Scikit-Learn, set `oob_score=True` when creating a BaggingClassifier to request an automatic oob evaluation after training (saved in `bag_clf.oob_score_`).

Sampling columns rather than rows

- The BaggingClassifier also lets you send a subset of features to each component model.
  - e.g., set `max_features=50` and `bootstrap_feature=True`
  - Useful for text data sets with lots of features.
  - Makes for a more diverse predictor, trading a bit more bias for lower variance.

Random Forests

- Now you know how random forests work:
  - Random Forests are optimized ensembles of decision trees with bagging.
- Good prediction performance – due to out-of-sample validation being baked in the training process.
- Also, interpretable because provides a feature importance ranking.

Random Forests

- The following code trains a Random Forest classifier with 500 trees (each limited to maximum 16 nodes), using all available CPU cores:

```
from sklearn.ensemble import RandomForestClassifier
rnd_clf = RandomForestClassifier(n_estimators=500,
                                 max_leaf_nodes=16,
                                 n_jobs=-1)

y_pred_rf = cross_val_predict(rnd_clf, X, Y)

confusion_matrix(Y,y_pred_rf)

feature_importances = forest_rnd_clf.feature_importances_
sorted(zip(feature_importances, features),reverse=True)
```

Gradient Boosted Machines and XGBoost

- A 2014 improvement to random forest is the gradient boosted machine.
  - the Python implementation XGBoost delivers state-of-the-art performance on structured data.
- Gradient boosting works by sequentially adding predictors to an ensemble – it fits the new predictor to the residual errors made by the previous predictor to gradually improve the model.

```
from xgboost import XGBRegressor, XGBClassifier, to_graphviz

xgb_clf = XGBClassifier()
cross_val_score(xgb_clf, X, Y).mean()

xgb_reg = XGBRegressor()
xgb_reg.fit(X,Y)
```

Feature Importance

- Random forests and boosted trees provide a metric of feature importance that summarizes how well each feature contributes to predictive accuracy.

```
sorted(zip(xgb_reg.feature_importances_, vocab), reverse=True)[:10]
from xgboost import plot_importance
plot_importance(xgb_reg, max_num_features=20)
```

Ensemble Application: Jelveh, Kogut, and Naidu (2016)

- This paper looks at political language and ideology in the economics literature.
- They use data on campaign contributions to assign a subset of economists to Republican or Democrat.
- Then they train a classifier to predict party based on the text of written articles
  - They use an ensemble PLS model, that "votes" in the same way as random forests, but the constituent voters are PLS regressors, rather than decision trees.
  - They control for topic choices using JEL K codes and LDA topics.
  - The model predicts with 70% accuracy.

JKN 2016: Results

- There is significant ideological sorting across fields:
  - law and economics is the most-right wing field, labor economics is the most left-wing field
- Right-wing economists report a higher labor supply elasticity than left-wing economists
- The ideology of editors does not affect ideology of published articles.