

Summary

Parallel Methods

Voting Classifiers

ال Voting classifier هو طريقة من خلالها بنعمل run لأكثر من model على التوازي (different models) ، وفي الآخر بناخذ ال prediction بتاع كل model وناخذ أكثر class كانت ال models ديه بتقول عليه بمعنى :

Model 1 say class 0

Model 2 say class 1

Model 3 say class 0

يبقا كده ال Majority of vote is class 0 .

بين على موديلز مختلفة .

Bagging and Pasting Classifier

بدل ما اعمل run ل models مختلفة ققدر هنا اعمل run لنفس ال model أكثر من مرة والاساس هنا هو ال DecisionTree لكن ققدر استخدم models تانية. وهنا كل model بي run على subset من الداتا مع تغير ال subset ده فى ال bootstrap aggregating which is Bagging ، او على نفس ال subset والى هو pasting . الميزة فى ال Bagging انه بيعمل Soft voting لو كان الموديل ليه probabilities .

بين على نفس الموديل .

Out-of-Bag Evaluation

فى ال Bagging بيحصل انى ال models بما انها بتعمل run على subset مختلفة من ال data ، هنا فى بعض ال models مش بتشوف باقى ال training data ، وهنا ققدر استخدم الجزء ده فى انى اعمل عليه evaluation بدل ما استخدم cross-val method .

Random Patches and Random Subspaces

ال Random patches لما اخلى ال instances وال features يختارو random عن طريق
bootstrap=True & bootstrap_features=True ، و Random_subspaces لما
Randomness ديه فى ال features فقط وهنا bootstrap_features=True &
bootstrap=False .

Random Forests

بدل من استخدام ensemble method عن طريق Bagging or pasting و تعمل ل pass
Random Forest ال decision Tree هو implementation ل ensemble method
باستخدام Decision Tree ، وفى معظم ال param بتاعت ال ensemble + Decision
Tree ، ال Random Forest اسرع من انك تعمل ل run ل Bagging or Pasting Classifier
باستخدام ال Decision Tree ، لانها بتختار احسن feature تعمل عنده split لل node من خلال
random subset من ال features مش كل ال features .

Extra Random Forests

بدل ما تختار best feature لا بتختار random threshold for each feature تعمل من
عنده split لل node وده بيكون اسرع بكثير .

Feature Importance

فيه ميزة قوية جدا فى ال Random Forest انك تقدر تحصل على اهم features بعد ما تعمل
train عن طريق ده تقدر تعمل select ليهم وتعمل traing من تانى وديه طريقة كويسة جدا لل
Features Selection .

**كل الى فات كنت تقدر تشغله على CPUs or server مختلفة على
التوازي .**

Sequential Methods

Boosting

الطرق الى فانت كان عن طريق انك بتعمل run لل models بس كل model على حدا وفي الآخر بتاخذ بال Voting.

هنا هبتدى انى اعمل run sequential for models ، كل واحد بيحاول انه يعمل correct للي قبله .

AdaBoost (Adaptive Boosting)

واحد من الطرق ديه هو AdaBoost ، وهو انى بروح اعمل run لاول موديل ، وبعد اعمل evaluate ليه على ال data الى عمل عليها train ، واشوف انهى instance حصل فيها Wrong classificatoin ، واروح اعمل updates لل weights المتعلقة بال instances ديه ، ومن ثم اروح اعمل run للموديل اللي بعديه ، وهكذا لحد ما اوصل للعدد ال انا محدده من ال models .

هنا بيحصل انه يعمل weights لكل instance على حد ونفس الكلام كل estimator بيكون عنده weights مختلفة .
بيرن على نفس الموديل .

Gradient Boosting

بدل ما اروح اعمل update لل weights بتاعت كل instance على حد ، لا هنا بروح اعمل updates لل weights بناء على ال error المتبقى ، بمعنى كل model بيروح يعمل predict على ال data الى عمل عليها training ، ثم ببتدى اشوف الفرق بين ال y_{actually} - $y_{\text{predicted}}$ ، ولما اروح اعمل fit للموديل الجديد اعمل passing للفرق ده .

```
tree_reg1 = DecisionTreeRegressor(max_depth=2)
tree_reg1.fit(X, y)
y2 = y - tree_reg1.predict(X)
tree_reg2 = DecisionTreeRegressor(max_depth=2)
tree_reg2.fit(X, y2)
```

Note we pass y2 the residual errors made by the previous predictor.

بيرن على نفس الموديل .