Introduction à l'Extraction de Connaissances

Chapitre IV

Part 4bis

Méthodes Ensemble

Alexandre Saidi
Ecole Centrale de Lyon
Département Mathématiques-Informatique
UMR LIRIS - CNRS

Novembre 2016

Texte sur RF et ses variantes

☞ Mettre ceci après RFs.
Site : http ://www.datasciencecentral.com/profiles/blogs/random-ized-forest-thought-vectors-to-build-a-new-class-of

Random-ized Forest : A new class of Ensemble algorithms
It's a known fact that bagging (an ensemble technique) works well on unstable algorithms like decision trees, artificial neural networks and not on stable algorithms like Naive Bayes.
The well known ensemble algorithm Random forest thrives on the ability of bagging technique which leverages the 'instability' of decisions trees, to help build a better classifier.
Even though, random forest attempts to handle the issues caused by highly correlated trees, does it completely solve the issue ? Can the decision trees be made more unstable than what random forest does, so that the learner be even more accurate ?
Lets talk about random forest, subsequently, I will introduce the concept of "Randomized forest" : for the lack of a better name :) :

Texte sur RF et ses variantes (suite)

1. Discards pruning : No more early stopping. If trees are sufficiently deep, they have very low bias.
Since, Mean Squared Error = $Variance + (Bias)^2$.
This explains why discarding pruning works for random forest.
2. The most important parameters to tune while building a random forest model are **mtry** i.e the number of variables per level and ntree i.e the number of tress to ensemble.
optimal 'mtry' can be estimated by using 'tuneRF'. tuneRF assumes the default value as the square root of total number of variables (lets say 'n') for classification problem, while n/3 for prediction problems.
It then calculates the out of bag error. Further, it goes for left and right estimation, assuming the 'mtry' to be equal to default value/step factor and (default value)* (step factor) respectively ; and calculates the out of bag error on both the scenarios and comes up the optimal mtry .
The step factor is manually provided,thus mtry definitely looks to be heavily dependent on the selected step factor, inappropriate assumption of which may

Texte sur RF et ses variantes (suite)

lead to misleading results. It is advised to keep it low so that more 'mtry' values are searched for.
If the step factor is anyway manually fed, which definitely restricts the search subspace ; can this be called an efficient optimization ?
➛ According to Adele Cutler, tuneRF does add bias and may lead to overfitting. !

3. Variable importance and the split criteria :
Information gain, entropy measure, ginni impurity is considered to be the metric for selecting the best attribute to split on.
Conventional recursive partitioning would consider all the variables together and search in the given space for the best attribute to split.
Random forest does it in random fashion by considered variables randomly by using the 'mtry' as the number of variables to use.
Can the selection of 'mtry' be randomized as well ? Can the splitting criteria be made more random and unstable by choosing one of the top-k variables based on information gain,instead of choosing the best only ?

Texte sur RF et ses variantes (suite)

Remember, adding instability worked well for bagging (e.g. random forest), adding more instability will increase the diversity we seek in an ensemble.
It's a false assumption or notion, that the variables which come out to be the most significant ones estimated by variable importance process ; are the ones most often used to split. It is essentially not true, in all the cases.
Can the variables with high variable importance be given more weightage while concluding which attribute to split on, along with the existing criterion ?

4. Correlation pruning in recursive fashion :
Highly correlated variables can cause multicollinearity, and deviation from orthogonal behavior ; which causes issues in classification problems.
To a great extent, multicollinearity is avoided even while dealing with correlated variables in case of random forest.
But what if the trees come out to be highly correlated ?
Can there be a pruning process by backward elimination in place to handle the correlated attributes and discard the correlated trees and not use them in final voting ?

Texte sur RF et ses variantes (suite)

5. Brewed/ composite features :
Learners are assumed to extract the **interaction** between attributes and the predictor.
Even highly advanced learners may miss out on the estimation of the interactions of different attributes together and subsequent interaction of these composite features to the predictor, be the deciding factor in choosing attribute as the best classifier and split .
Brewed features may help in providing the information in a tree, which may otherwise be left out when growing trees.
Composite features combined with the organic ones, during the random space search for splits can add more randomness and diversity to the ensemble built using the votes by the isolated learners.
These are just the 'thought vectors', the 'geometry' is soon to be published :)

Suite

• Site http ://www.datasciencecentral.com/profiles/blogs/how-and-why-decorrelate-time-series
et (le site parent) http ://www.datasciencecentral.com/profiles/blogs/21-great-articles-and-tutorials-on-time-series
Donne des trucs sur les times series.

• J'au asuui sauvegardé le site
http ://www.datasciencecentral.com/profiles/blogs/difference-between-machine-learning-data-science-ai-deep-learning
Qui n'est pas mal.

Table des matières