Random Forest

Fabrício Barth

Agosto de 2020

Ensemble Learning

- Métodos que geram diversos modelos e agregam o seu resultado.
- No caso do Random Forest, são geradas diversas árvores e cada árvore é gerada considerando apenas um sub-conjunto do conjunto de treinamento.
- Este tipo de algoritmo também é chamado de *Bootstrap Aggregating* ou **Bagging**.

Random Forest

- O algoritmo possui apenas dois parâmetros configuráveis:
  - ⋆ quantidade de atributos considerados em cada árvore (m_{try}), e;
  - ⋆ quantidade de árvores (n_{tree}).

Random Forest

Para problemas de classificação e regressão o algoritmo funciona da seguinte forma:

- Cria n_{tree} sub-conjuntos de exemplos a partir do dataset original.
- Para cada sub-conjunto de exemplos cria-se uma árvore de classificação ou regressão sem poda. A criação de cada árvore considera apenas um sub-conjunto de exemplos: m_{try} atributos selecionados aleatoriamente e $2/3$ dos exemplos também selecionados aleatoriamente.

- A predição para novos dados acontece pela agregação das predições das n_{tree} árvores.
- Para problemas de **classificação** é considerado a maioria dos votos.
- Para problemas de **regressão** é considerado a média dos votos.

Particularidades de implementação no sklearn

```
max_features : int, float, string or None, optional (default="auto")
The number of features to consider when looking for the best split:

If int, then consider max_features features at each split.
If float, then max_features is a fraction and int(max_features * n_features)
If "auto", then max_features=sqrt(n_features).
If "sqrt", then max_features=sqrt(n_features) (same as ?auto?).
If "log2", then max_features=log2(n_features).
If None, then max_features=n_features.
```

```
max_depth : integer or None, optional (default=None)
The maximum depth of the tree. If None, then nodes are expanded
until all leaves are pure or until all leaves contain less than
min_samples_split samples.

warm_start : bool, optional (default=False)
When set to True, reuse the solution of the previous call to fit
and add more estimators to the ensemble, otherwise, just fit a
whole new forest.
```

Estimativa de erro

- Uma estimativa de erro, usando apenas o conjunto de treinamento, pode ser obtida através do conjunto de treinamento. Ao invés de ser utilizado algum outro método, como *cross-validation*.
- Para cada árvore construída é usado um sub-conjunto de exemplos. $1/3$ dos exemplos são mantidos fora do conjunto de treinamento. Estes exemplos mantidos fora do conjunto de treinamento são utilizados como teste.

Exemplos

https://github.com/fbarth/ml-espm/blob/master/scripts/python/05_01_random_forest.ipynb

Hiperparâmetros

Um modelo de Random Forest tem os seguintes hiperparâmetros:

- n_estimators = número de árvores na floresta.
- max_features = número máximo de atributos considerados na seleção de um atributo.
- max_depth = número máximo de níveis em cada árvore de decisão.

- min_samples_split = número mínimo de exemplos que devem ser considerados antes de cada divisão de nodo.
- min_samples_leaf = número mínimo de exemplos em cada nodo final.
- bootstrap = método para amostragem de exemplos (com ou sem *replacement*). O default é $True$, ou seja, com *replacement*.

GridSearch

```
from sklearn.model_selection import GridSearchCV
param_grid = {
 'n_estimators': [100, 200, 500, 600, 800, 1000],
 'max_features': ['auto', 'sqrt', 'log2'],
 'max_depth' : [5,10,50,100,150,None]
}
rfc=RandomForestClassifier(random_state=4)
CV_rfc = GridSearchCV(
 estimator=rfc, param_grid=param_grid,
 cv= 3, verbose=1, n_jobs=4)
CV_rfc.fit(X_train, y_train)
```

GridSearch

- Executa todas as combinações considerando todos os valores de todas as variáveis do *grid*.
- No caso do exemplo anterior, $6 \times 3 \times 6$, que gera 108 modelos possíveis.
- Além disso, cada modelo é gerado 3 vezes por que o *GridSearchCV* executa uma rotina de cross-validation igual a 3.

GridSearch

```
from sklearn.model_selection import GridSearchCV
param_grid = {
    'n_estimators': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000],
    'max_features': ['auto','sqrt', 'log2'],
    'max_depth' : [5,10,50,100,150,None]
}
rfc=RandomForestClassifier(random_state=4)
CV_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv= 3, verbose=1, n_jobs=4)
CV_rfc.fit(X_train, y_train)
```

```
Fitting 3 folds for each of 180 candidates, totalling 540 fits
[Parallel(n_jobs=4)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=4)]: Done  42 tasks      | elapsed:  1.3min
[Parallel(n_jobs=4)]: Done 192 tasks      | elapsed:  7.1min
[Parallel(n_jobs=4)]: Done 442 tasks      | elapsed: 20.8min
[Parallel(n_jobs=4)]: Done 540 out of 540 | elapsed: 25.9min finished
CPU times: user 11.1 s, sys: 164 ms, total: 11.2 s
Wall time: 26min 3s
```

GridSearch e Random SearchCV

- Apesar do processo demorar, o GridSearch retorna a melhor configuração para os parâmetros testados.
- Ao invés de testar todas as possibilidades, pode-se utilizar o Random SearchCV para testar apenas parte das configurações de forma aleatória.
- o Random SearchCV testa no máximo n combinações, onde n é determinado pelo parâmetro n_iter.

Exemplo de código

https://github.com/fbarth/ml-espm/blob/master/scripts/python/05_02_random_forest.ipynb

Implementação de RandomForest para problemas desbalanceados

- imblearn.ensemble.BalancedRandomForestClassifier
- Esta implementação faz under-sample para balancear o conjunto de treinamento em cada seleção de exemplos para construção de cada uma das árvores da floresta.

Considerações finais

- Em comparação com as árvores de decisão, o que se perde é a estrutura simples e interpretável; o que se ganha é a redução de *overfitting*.

Material de **consulta**

- Breiman and Cutler. Random Forests. Acessado em https://www.stat.berkeley.edu/b̃reiman/RandomForests/
- Liaw and Wiener. Classification and Regression by randomForest. R News 2 (3): 18–22 (2002)
- https://scikit-learn.org/stable/modules/ensemble.html#forest

- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
- http://rpubs.com/fbarth/exemploRandomForest