Ensemble models: Bagging and Boosting

Fabrício Barth

Setembro de 2020

Ensemble Learning

- Métodos que geram diversos modelos e agregam o seu resultado.
- Basicamente, existem dois tipos:

- ⋆ **Bagging**: cria diversos modelos de forma independente e ao realizar uma predição faz uso da média da opinião de todos os modelos.
- ⋆ **Boosting**: cria um modelo robusto a partir da criação e aperfeiçoamento de diversos modelos fracos. Um modelo fraco é inicialmente criado e então o próximo modelo é criado tentando resolver os erros do primeiro modelo e assim sucessivamente.

Algoritmos do tipo Bagging

- from sklearn.ensemble import RandomForestClassifier
- from imblearn.ensemble import BalancedRandomForestClassifier

Algoritmos do tipo Boosting

- AdaBoost
- Gradient Boost
- XGBoost

AdaBoost

- Usa um conjunto de stumps (árvores com só um nível de decisão - um nodo e duas folhas) como modelo. Ou seja, utiliza diversos modelos "fracos" para compor um modelo "forte".
- No entanto, cada stump depende do stump anterior. Os erros que o primeiro stump comete influencia como o segundo stump é feito e assim sucessivamente.
- Alguns stumps tem peso superior que outros stumps.

- Este é um vídeo que explica de forma muito didática como o algoritmo Adaboost funciona:
  https://www.youtube.com/watch?v=LsK-xG1cLYA
- Esta é uma aula do professor Patrick Winston no curso de Inteligência Artificial do MIT:
  https://www.youtube.com/watch?v=UHBmv7qCey4
  Explica de uma maneira mais formal como a ideia de Boosting funciona.

Gradient Boost para problemas de regressão

- A forma de funcionamento do Gradient Boost é muito parecida com a forma de funcionamento do AdaBoost.
- No entanto, ao invés de criar vários stumps, o Gradient Boost cria diversas árvores com profundidade limitada. Até 4, 8 ou 32, por exemplo.
- A única diferença está no primeiro modelo que é apenas um nodo. Em problemas de regressão é a média do valor a ser predito. Em problemas de classificação é a maioria dos valores no conjunto de treinamento.

- Cada árvore também depende do erro da árvore anterior, assim como no AdaBoost cada stump depende do stump anterior.
- Em um problema de regressão, cada árvore calcula o valor pseudo residual[a] (o erro) do modelo anterior.
- O algoritmo Gradient Boost faz uso de um *Learning rate* (uma valor entre 0 e 1) para fazer com que o algoritmo possa ir aos poucos para a direção certa, ou seja, para o valor predito certo.

[a] para diferenciar do conceito de resíduo de uma regressão linear

- Outro vídeo que explica de forma muito didática como o algoritmo em questão funciona:
  https://www.youtube.com/watch?v=3CC4N4z3GJc

Referências

- https://scikit-learn.org/stable/modules/ensemble.html#adaboost
- AdaBoost: https://www.youtube.com/watch?v=LsK-xG1cLYA
- Learning: Boosting. MIT 6.034 Artificial Intelligence, Fall 2010. Instructor: Patrick Winston. https://www.youtube.com/watch?v=UHBmv7qCey4

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)
```