AutoML

Automating the hyperparameter search

Lyle Ungar

Auto-SKlearn
Ensemble selection

Auto-ML

- Automate the search for hyperparameters and network architectures in a neural net

- Learn best hyperparameters in an ensemble of sk-learn models

Auto-Sklearn

- 15 Classifiers
- 14 feature preprocessing methods
- 4 data preprocessing methods

→ 110 hyperparameters

Combined Algorithm Selection and Hyperparameter (CASH) Optimization

Preprocessing & Methods

extreml. rand. trees prepr. feature selection
fast ICA
feature agglomeration
kernel PCA
rand. kitchen sinks (random projection)
linear SVM prepr. L1 feature selection
no preprocessing
nystroem sampler (random projection)
PCA
polynomial
random trees embed.
select percentile
select rates

---

one-hot encoding
imputation
balancing
rescaling

AdaBoost (AB)
Bernoulli naïve Bayes
decision tree (DT)
extreml. rand. trees
Gaussian naïve Bayes
gradient boosting (GB)
kNN
LDA Linear Discriminant analysis
linear SVM
kernel SVM
multinomial naïve Bayes
passive aggressive
QDA Quadratic Discriminant analysis
random forest (RF)
Linear Class. (SGD)

Auto-Sklearn

- **Warmstart/Metalearning:** Start from hyperparameters that worked in the past for similar datasets.
  - Based on 38 metafeatures of 140 datasets
- Uses **Bayesian optimization**
  - Fit a random forest model predicting performance from hyperparameters and use it to find the optimum
    - speed up by discarding values that look bad on the first fold of 10-fold CV
- Use **Ensemble** of the 50 best classifiers considered

Ensemble selection

- **Greedy (stagewise)**
- **Start from an empty ensemble**
- **Iteratively add the model that minimizes ensemble validation loss**
  - with uniform weight, but allowing for repetitions
- **Why not optimize the weights on each model?**

Metafeatures

- **Number of features & observations**
  - With transformations
  - Number and percentage missing
  - Number real or categorical
- **Class probability stats**
  - Min, max, entropy…

Auto-Sklearn is solid

- Performance (with limited CPU) was third among a large set of human competitors

Better preprocessing

- Use good automatic encoding of categorical variables
  - E.g. "CatBoost" encoding
  - Replace each category with a “target statistic” such as the expected value of *y* for that category
  - Need to be clever to avoid overfitting (“leakage”)
  - See https://catboost.ai/docs/concepts/algorithm-main-stages_cat-to-numberic.html
  - https://contrib.scikit-learn.org/categorical-encoding/ has lots of categorical encodings, including CatBoost

AutoML using Metadata Language Embeddings

- **Use text description of problems to pick hyper-parameters**
  - Use vector embeddings of dataset title, description and keywords
  - For each new dataset, find most similar prior dataset and use its hyperparameters
  - The similarity metric is learned (supervised)

Drori et al 2019

Conclusions

- **AutoML is close to the best humans**
  - And less likely to overfit
  - Different ensembles for different problem types
- **To really avoid overfitting, do nested CV**
  - For each of ten folds, on the 90%
    - Do 10-fold CV to find the best method
  - Observe performance on the held-out 10%