Exhaustive Tuning

Use exhaustive grid search techniques for hyperparameter tuning.

Chapter Goals:

- Learn how to use grid search cross-validation for exhaustive hyperparameter tuning

A. Grid-search cross-validation

If our application requires us to absolutely obtain the best hyperparameters of a model, and if the dataset is small enough, we can apply an exhaustive grid search for tuning hyperparameters. For the grid search cross-validation, we specify possible values for each hyperparameter, and then the search will go through each possible combination of the hyperparameters and return the model with the best combination.

We implement grid search cross-validation with the `GridSearchCV` object (part of the `model_selection` module).

```
reg = linear_model.BayesianRidge()
params = {
  'alpha_1':[0.1,0.2,0.3],
  'alpha_2':[0.1,0.2,0.3]
}
reg_cv = GridSearchCV(reg, params, cv=5, iid=False)
# predefined train and test sets
reg_cv.fit(train_data, train_labels)
print(reg_cv.best_params_)
```

RUN SAVE RESET

In the code example above, we searched through each possible pair of α_1 and α_2 values based on the two lists in the `params` dictionary. The search resulted in an α_1 value of 0.3 and an α_2 value of 0.1. For each of the models we've covered, you can take a look at their respective scikit-learn code documentation pages to determine the model's hyperparameters that can be used as the `params` argument for `GridSearchCV`.

The `cv` keyword argument represents the number of folds used in the K-Fold cross-validation for grid search. The `iid` keyword argument relates to how the cross-validation score is calculated. We use `False` to match the standard definition of cross-validation. Note that in later updates of scikit-learn, the `iid` argument will be removed from `GridSearchCV`.

Since exhaustive grid search performs cross-validation on each possible hyperparameter value combination, it can be incredibly slow for larger datasets. It should only be used if the dataset is reasonably small and it is important to choose the best hyperparameter combination.