

Hyperparameter search

Taehoon Ko (taehoonko@snu.ac.kr)

목표

- 하이퍼파라미터(Hyperparameter)의 의미를 이해한다.
- 하이퍼파라미터를 탐색하기 위한 방법을 이해한다.
 - Grid search
 - Randomized search

Hyperparameter search

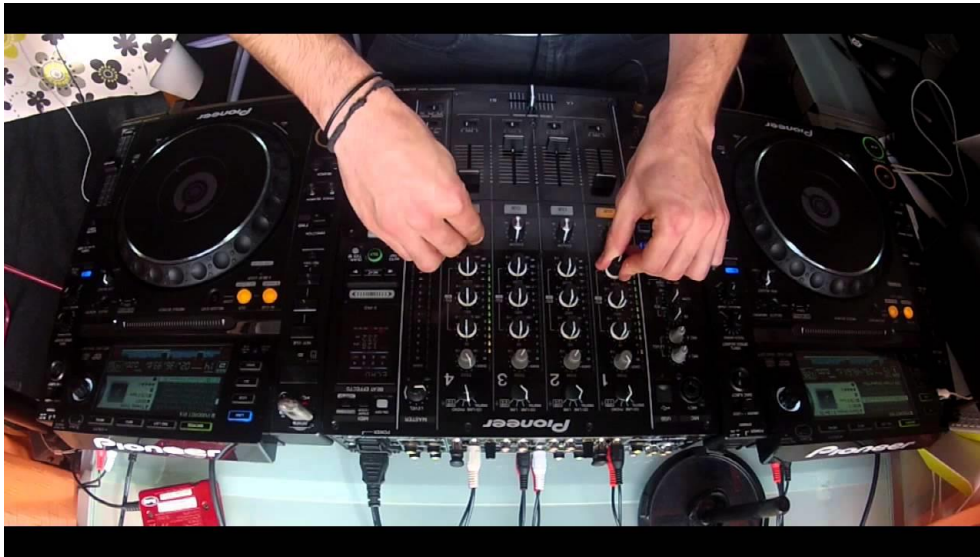
Hyperparameter control

Hyperparameter tuning

Hyperparameter optimization

Club DJ, Pilot → Data scientist

- DJ set, airplane → machine learning model
- Buttons, dials → Hyperparameters for the model



Control parameters of the model

- Example: Ridge

`sklearn.linear_model.Ridge`

```
class sklearn.linear_model.Ridge(alpha=1.0, fit_intercept=True, normalize=False, copy_X=True, max_iter=None, tol=0.001, solver='auto', random_state=None)
```

[\[source\]](#)

Linear least squares with L2 regularization.

Initial parameters

This model solves a regression model where the loss function is the linear least squares function and regularization is given by the L2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression (i.e., when `y` is a 2d-array of shape `[n_samples, n_targets]`).

Control parameters of the model

- Example: MLPClassifier

`sklearn.neural_network.MLPClassifier` ¶

```
class sklearn.neural_network.MLPClassifier(hidden_layer_sizes=(100, ), activation='relu',  
solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001,  
power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False,  
warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False,  
validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10)
```

[\[source\]](#)

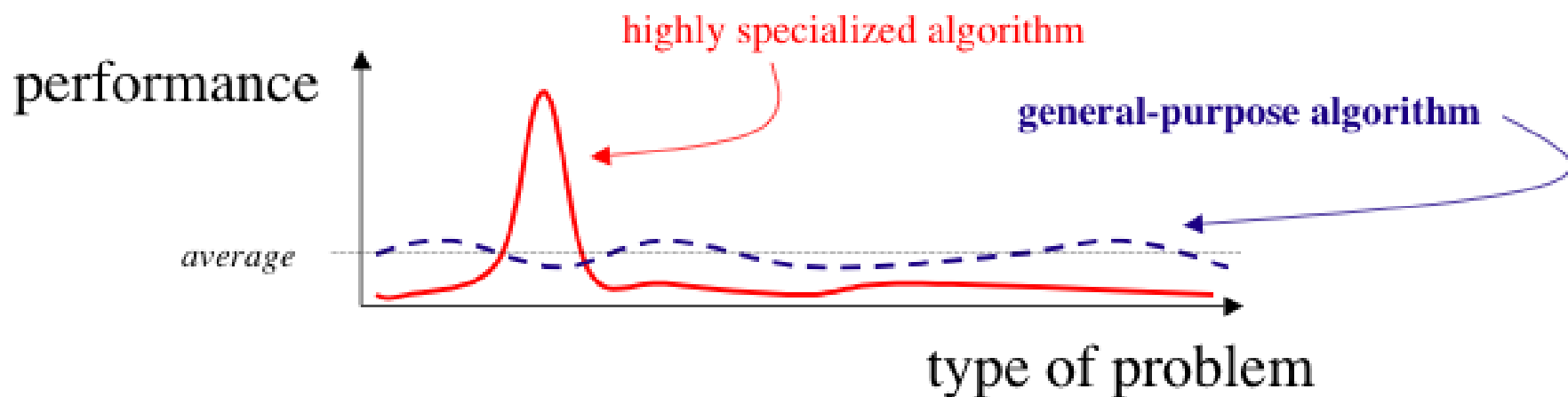
Initial parameters

Hyperparameters are very important!

- All machine learning models have hyperparameters.
- If you want to get good models, **you have to control hyperparameters!**
- Sadly, there is no perfect way to find optimal hyperparameters.

No Free Lunch (NFL) theorem

- We have dubbed the associated results NFL theorems because they demonstrate that if an algorithm performs well on a certain class of problems then it necessarily **pays for that with degraded performance on the set of all remaining**



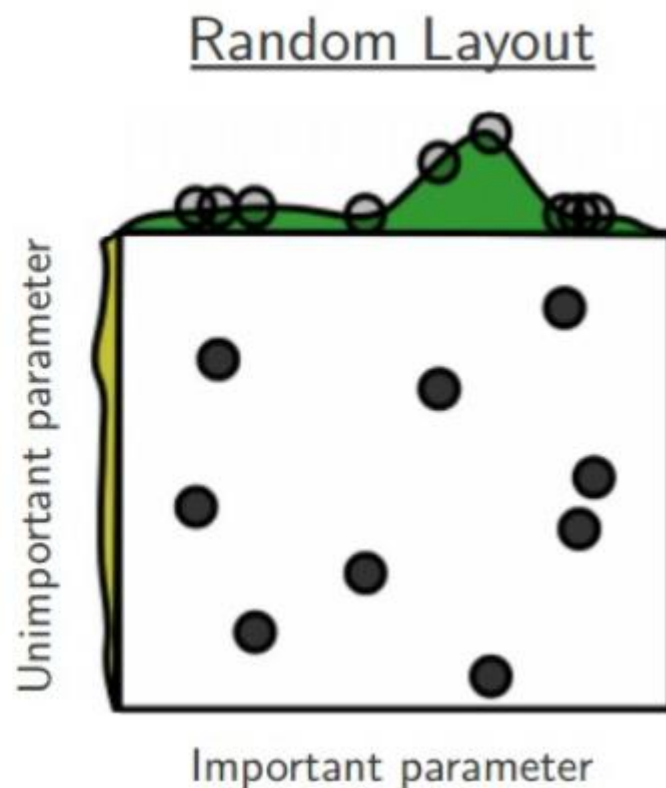
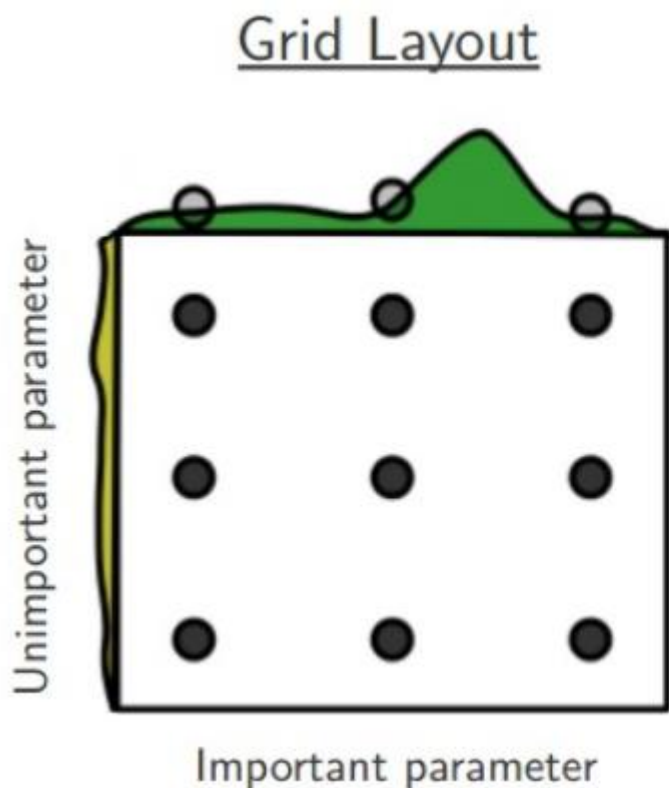
http://ecmendenhall.blogspot.com/2006_02_10_archive.html

Trial and Error

- 여러 하이퍼파라미터를 이용해 모델을 학습
 - 여러 하이퍼파라미터 후보들을 이용
 - 많이 탐색할수록 좋은 결과를 얻을 수 있겠지만, 시간이 오래 소요될 것
 - Training set를 이용해 학습
- 학습된 모델을 평가 ➔ 최적의 하이퍼파라미터를 결정
 - Validation set (a.k.a. development set) 을 이용하여 학습된 모델을 평가

Grid search vs. Randomized search

- 두 방법 모두 trial and error를 기본으로 함
- Grid search는 후보값을, Randomized search는 분포를 제시
- 두 방법 모두 Parallel processing이 가능함



하이퍼파라미터 탐색 시 유의할 점

- 하이퍼파라미터가 알고리즘에서 어떤 역할을 하는지 정확히 파악하는 것이 필요함.
 - Ex1) Ridge, Lasso 학습 시 'alpha'는 어떤 후보값 (혹은 어떤 분포) 을 주어야 하는가?
 - Ex2) MLPClassifier 학습 시 'learning_rate_init'에는 어떤 후보값 (혹은 어떤 분포) 을 주어야 하는가?
- 공부가 필요함.
- 경험치는 많은 도움이 됨. (~~사람이 trial and error로 학습되는 효과~~)

(참고) 또 다른 탐색 방법

- Bayesian optimization

- 하이퍼파라미터를 맵핑하는 확률적 모델 (probabilistic model) 설계
 - Validation set에 대한 평가를 활용
- GP EI MCMC라는 방법 제안 → Grid search, Randomized search보다 더 최적값을 빠르게 찾는 것을 실험적으로 평가
 - EI: Expected Improvement / MCMC: Markov Chain Monte Carlo

- Using evolutionary algorithms

- Ex) 유전체 알고리즘 (Genetic Algorithm)
- 좋은 결과가 나온 하이퍼파라미터 집합 후보군을 섞어서 더 좋은 하이퍼파라미터 집합을 계속 찾아나가는 방법