

Aprendizado de Máquina

Aula 5.1 - Tuning de hiperparâmetros


Adriano Rivolli

rivolli@utfpr.edu.br

Especialização em Inteligência Artificial

Universidade Tecnológica Federal do Paraná (UTFPR)
Câmpus Cornélio Procópio
Departamento de Computação

Conteúdo

- 
- 1** Visão Geral
 - 2** Abordagens
 - 3** Metodologia

Visão Geral

Por que fazer tuning?

- Para melhorar o desempenho em tarefas específicas
 - ▶ Hiperparâmetros são configurações externas que afetam na escolha do melhor modelo
 - ▶ Alguns algoritmos são mais sensíveis as escolhas dos hiperparâmetros
- Diferentes conjuntos de dados requerem diferentes configurações
- A técnica de tentativa e erro é laboriosa e pouco precisa

Abordagens de Tuning

- Busca aleatória (*Random search*)
- Busca em grade (*Grid search*)
- Otimizações
 - ▶ Gradient-based optimization
 - ▶ Bayesian Optimization
 - ▶ Meta-heurística

Dicas gerais

- Função objetivo
 - ▶ Escolha uma ou múltiplas métricas para otimizar
- Amplie o espaço de busca
 - ▶ Considere a avaliação de todo os *pipeline*
- Desenvolvimento e avaliação
 - ▶ Separe o conjunto de validação e de teste
- Paralelize as tarefas
 - ▶ Poupe tempo
- Previna-se das falhas
 - ▶ Salve resultados intermediários

Abordagens

Busca aleatória

- Consiste na busca aleatória dos valores dos hiperparâmetros
- Requer que a especificação do espaço de busca
- Consiste em uma abordagem exploratória e não sistemática
- O número de modelos que serão avaliados é definido pelo usuário

Busca em grade

- Consiste em especificar uma grade de valores dos hiperparâmetros que serão analisados
- A quantidade de modelos avaliados é dado pela multiplicação dos números de valores considerado na grade
 - ▶ $C = [0.001, 0.01, 0.1, 1, 10, 100, 1000]$
 - ▶ $\text{Gamma} = [0.01, 0.1, 1, 10, 100]$
 - ▶ $N. \text{ Modelos} = 7 * 3 = 35$
- Abordagem sistemática que avalia todas as combinações
- Mais custosa computacionalmente

Exemplo da busca em grade

■ $C = [0.001, 0.01, 0.1, 1, 10, 100, 1000]$

■ $\text{Gamma} = [0.01, 0.1, 1, 10, 100]$

▶ $C = 0.001, \text{Gamma} = 0.01$

▶ $C = 0.001, \text{Gamma} = 0.1$

▶ $C = 0.001, \text{Gamma} = 1$

▶ $C = 0.001, \text{Gamma} = 10$

▶ $C = 0.001, \text{Gamma} = 100$

▶ $C = 0.01, \text{Gamma} = 0.01$

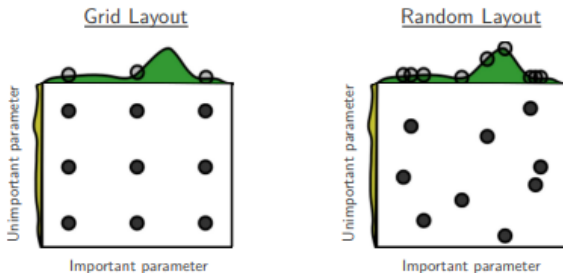
▶ ...

▶ $C=1000, \text{Gamma} = 100$

Múltiplas grades


- Alguns hyperparâmetros são condicionais a outros
- Neste caso, múltiplas grades devem ser combinadas
- Exemplo:
 - ▶ No SVM algumas funções de kernels podem ter hyperparâmetros específicos
 - ▶ Ao combinar diferentes kernels, faz sentido testar valores diferentes para cada

Comparação: Grid x Random



Fonte: Bergstra, J., Bengio, Y. 2012. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research.

Outras abordagens

- 
- Utilizam resultados das interações prévias para escolher os próximos valores candidatos
 - Em alguns casos, isso impede a possibilidade de paralelismo
 - Normalmente o processo para em um critério de convergência
 - Cada abordagem usa uma forma distinta de explorar o caminho a ser percorrido

Metodologia

Processo de validação

- Treino, validação e teste
- Validação cruzada aninhada

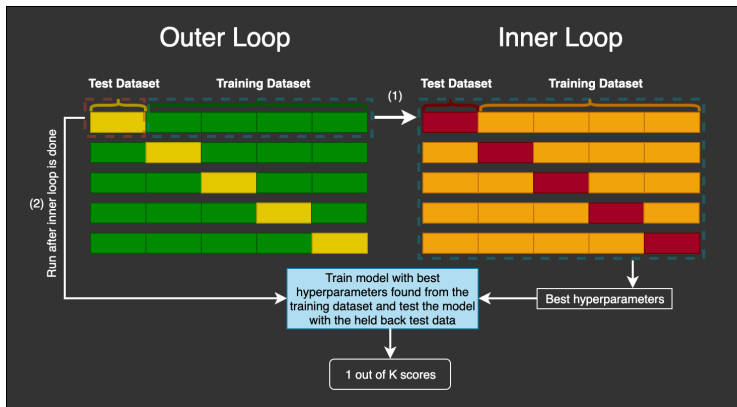
Treino, validação e teste

- O conjunto de **treino** é utilizado para o treinamento
 - ▶ Um modelo para cada combinação dos hiperparâmetros
- O conjunto de **validação** é utilizado para avaliar os modelos
 - ▶ O melhor modelo é escolhido a partir desta avaliação
 - ▶ Modelo final:
 - Fazer um comitê com os X melhores modelos
 - Treinar um novo modelo com todos os dados
- O conjunto de **teste** é utilizado para avaliar o modelo final


Validação cruzada aninhada

- Para cada partição de treinamento, uma nova validação cruzada é usada para encontrar os valores dos hiperparâmetros
 - ▶ **outer-loop** é a validação cruzada externa, é utilizada para avaliar o modelo final
 - ▶ **inner-loop** é a validação cruzada interna, é utilizada para a escolha dos valores dos hiperparâmetros

Validação cruzada aninhada (ilustração)



Modelo final

- 
- O modelo relativo ao resultado da melhor partição
 - Comitê com todos as partições
 - Induzir um modelo com todos os dados usando os melhores valores dos hyperparâmetros