

Objetivo da atividade

Objetivo desta atividade é aplicar o algoritmo de Otimização de Enxame de Partículas (*Particle Swarm Optimization - PSO*) para resolver uma sub-tarefa do nosso *pipeline* de ciência de dados.

1 O que fazer?

Nessa atividade você pode optar por uma entre duas tarefas diferentes envolvendo o uso do PSO:

- realizar o ajuste de hiperparâmetros de um classificador (*hyperparameter tuning*); ou
- realizar seleção automatizada de atributos (*feature selection*).

1.1 Ajuste de Hiperparâmetros

Se a tarefa escolhida for o ajuste de hiperparâmetros:

- escolha um algoritmo para ter seus hiperparâmetros ajustados. **Esse algoritmo deve ter pelo menos 3 hiperparâmetros ajustáveis** (dimensões do problema);
- escolha um dataset, pode ser classificação ou regressão. Evite os datasets convencionais (iris, wine, glass, digits), e os datasets já usados nos notebooks da disciplina.
- Divida o dataset escolhido em partições de treino e teste. Divida os dados usando *holdout* na proporção 2/3 (treino) e 1/3 (teste).
- aplique o PSO para otimizar os hiperparâmetros do algoritmo usando o conjunto de treinamento;
- avalie os melhores hiperparâmetros encontrados pelo PSO no conjunto de teste;
- Repita o processo 5x, variando a amostragem na divisão treino/teste (. Os valores de hiperparâmetros encontrados são iguais?
- compare os resultados obtidos com a performance do mesmo algoritmo usando os seus hiperparâmetros default (sem ajuste) nas mesmas práticas de dados;
- faça experimentos com diferentes configurações do PSO, variando tamanho da população e número de iterações.

1.2 Seleção Automatizada de Atributos

Se a tarefa escolhida for a seleção de atributos:

- escolha um algoritmo de aprendizado de máquina;
- escolha um dataset, pode ser classificação ou regressão. Evite os datasets convencionais (iris, wine, glass, digits), e os datasets já usados nos notebooks da disciplina. **Priorize datasets com alta dimensionalidade, isto é, um número elevado de atributos (>50).**
- Divida o dataset escolhido em partições de treino e teste. Divida os dados usando *holdout* na proporção 2/3 (treino) e 1/3 (teste).

-
- aplique o PSO para encontrar o melhor subconjunto de atributos do dataset usando a partição de treinamento e o algoritmo selecionado;
 - avalie o subconjunto de atributos encontrado pelo PSO na partição de teste. Quantos e quais atributos foram retornados pela otimização?
 - Repita o processo 5x, variando a amostragem na divisão treino/teste. Os mesmos atributos foram retornados nas 5 execuções?
 - compare os resultados obtidos com a performance do mesmo algoritmo usando o dataset original (com todas as features);
 - faça experimentos com diferentes configurações do PSO, variando tamanho da população e número de iterações.

1.3 Relatório com análises experimentais

Independente da tarefa escolhida, elabore um relatório técnico (documento pequeno, .pdf, ou notebook) com os experimentos realizados. Explique as etapas/experimentos realizados e justifique suas escolhas. Apresente os resultados, de preferência por meio de figuras/gráficos/tabelas, e descreva conclusões críticas com base nestes resultados observados.

2 Opcinal: o que também pode fazer?

Se você conseguiu realizar uma das duas tarefas acima e ainda tem algum tempo para dedicar aos estudos:

- Implemente a sua versão do PSO;
- compare seu algoritmo com as implementações existentes;
- quão bem seus códigos foram em relação as ferramentas da literatura? Como seu código poderia ser futuramente melhorado?

3 Alguns links que podem ser úteis

- **mlr3** - *Machine Learning in R: framework* em R para implementação de soluções de Aprendizado de Máquina. Ele é bem completo, quase tudo já está implementado, o problema pode ser entender como encaixar cada peça. Link: <https://mlr3.mlr-org.com>;
- **pso** - *PSO in R*: implementação do PSO no R. Link: <https://cran.r-project.org/web/packages/pso/pso.pdf>
- Exemplo de uso do PSO com R: <https://rpubs.com/Argaadya/intro-PSO>
- **scikit learn** - *Machine Learning in Python* - contapartida do mlr mas em Python. <https://scikit-learn.org/stable/>
- **pyswarms**: implementação do PSo no Python. Link: <https://pypi.org/project/pyswarms/>

Outros links:

- https://pyswarms.readthedocs.io/en/latest/examples/usecases/train_neural_network.html
- https://pyswarms.readthedocs.io/en/development/examples/feature_subset_selection.html
- <https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/>
- <https://www.openml.org>
- <https://archive.ics.uci.edu/ml/datasets.php>