

Aprendizado Descritivo

Tutorial: Como executar MCTS4DM

GRUPO HACKER

ADRIANO VICTOR MAGALHÃES ZANINI

BRUNO LEAL FONSECA

INDRA MATSIENDRA CARDOSO DIAS RIBEIRO

JOSÉ MARTINS DA ROSA JUNIOR

LEONARDO DE OLIVEIRA MAIA

MATHEUS VAZ LEAL LIMA

VITÓRIA MARIA SILVA BISPO

¹ Departamento de Ciência da Computação

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

1. Instalação do Java

Para executar o programa MCTS4DM, é necessário instalar o Java Development Kit (JDK) e o Java Runtime Environment (JRE) na versão 22. Siga os passos abaixo para realizar a instalação:

1.1. Baixar e instalar o JDK

- Acesse o site oficial da Oracle: [Oracle JDK 22](#).
- Escolha a versão adequada, indicamos o JDK 22.0.1, para o seu sistema operacional (Windows, macOS ou Linux) e faça o download do instalador.
- Execute o instalador baixado e siga as instruções na tela para concluir a instalação.

1.2. Configurar a variável de ambiente JAVA_HOME

- No Windows, abra o Painel de Controle e vá em "Sistema e Segurança" > "Sistema" > "Configurações avançadas do sistema" > "Variáveis de ambiente".
- Crie uma nova variável de sistema chamada `JAVA_HOME` e defina seu valor para o caminho onde o JDK foi instalado, por exemplo, `C:\Program Files\Java\jdk-20`.
- Adicione `%JAVA_HOME%\bin` ao `PATH` nas variáveis de sistema.

1.3. Verificar a instalação

- Abra o terminal (Prompt de Comando no Windows, Terminal no macOS ou Linux) e digite `java -version` e `javac -version` para verificar se o Java foi instalado corretamente.

2. Clonar o Repositório

Para obter o código fonte do MCTS4DM, você precisa clonar o repositório do GitHub. Siga os passos abaixo para clonar o repositório em diferentes sistemas operacionais:

2.1. Windows

- Baixe e instale o Git a partir do site oficial: [Git para Windows](#).
- Após a instalação, abra o Git Bash.
- No Git Bash, execute o seguinte comando para clonar o repositório:

```
git clone https://github.com/guillaume-bosc/MCTS4DM.git
```

2.2. macOS

- Abra o Terminal.
- Instale o Git usando o Homebrew com o comando:

```
brew install git
```

- No Terminal, execute o comando para clonar o repositório:

```
git clone https://github.com/guillaume-bosc/MCTS4DM.git
```

2.3. Linux

- Abra o Terminal.
- Instale o Git usando o gerenciador de pacotes da sua distribuição Linux, por exemplo, no Ubuntu:

```
sudo apt-get install git
```

- No Terminal, execute o comando para clonar o repositório:

```
git clone https://github.com/guillaume-bosc/MCTS4DM.git
```

3. Executar o MCTS4DM

Depois de configurar o ambiente Java, é hora de executar o MCTS4DM. Primeiro, certifique-se de que os arquivos de dataset e o arquivo de configuração de parâmetros estejam preparados. Em seguida, siga os passos abaixo:

3.1. Preparar os arquivos de dataset e configuração

- Navegue até a pasta `datasets` no diretório onde o programa `MCTS4DM` está localizado.
- Dentro da pasta `datasets`, crie uma pasta exclusiva para o dataset, por exemplo, `BreastCancer`.
- Coloque os arquivos `properties.csv` e `qualities.csv` dentro da pasta `BreastCancer`, por exemplo.
- Certifique-se de que o arquivo `parameters.conf` esteja configurado corretamente, apontando para os caminhos relativos dos arquivos de dataset, para isto, o arquivo padrão do repositório `parameters.conf` explica cada parâmetro e como configurar.

3.2. Formato dos arquivos de dataset

- `properties.csv`: Contém a tabela de atributos, onde cada linha é um objeto e cada coluna é um atributo. O separador entre colunas deve ser `"\t"`.
- `qualities.csv`: Contém a tabela de rótulos, onde cada linha é um objeto (na mesma ordem do `properties.csv`) e cada coluna é um rótulo de alvo.

3.3. Executar o MCTS4DM

- No terminal, navegue até a pasta onde está o arquivo `MCTS4DM.jar`.
- Execute o comando:

```
java -jar MCTS4DM.jar <caminho para o arquivo de  
configura o >
```

- Por exemplo:

```
java -jar MCTS4DM.jar parameters.conf
```

4. Resultados da Execução

Após a execução do programa, os resultados estarão disponíveis na pasta especificada no parâmetro `resultFolderName` dentro da pasta `results`. Quatro arquivos serão gerados:

- `info.log`: Contém os valores dos parâmetros utilizados, o tempo de execução e a data da execução.

- **results.log:** Cada linha contém um subgrupo no formato `<Descrição>\t<Alvos>\t<Medida>\t<E11>\t<E10>\t<E01>`. Onde E11 é o número de objetos que respeitam a descrição e estão associados aos alvos, E10 é o número de objetos que respeitam a descrição mas não estão associados aos alvos, e E01 é o número de objetos que não respeitam a descrição mas estão associados aos alvos.
- **support.log:** Contém o suporte de cada subgrupo, onde cada linha corresponde ao suporte do subgrupo correspondente no arquivo **results.log**.
- **supportE11.log:** Contém apenas os IDs dos objetos em E11.

5. Geração de Plots com Base nos Resultados

Para gerar plots dos resultados obtidos, siga os passos abaixo:

5.1. Acessar a pasta `GenerateDataPlot`

- Navegue até a pasta `GenerateDataPlot` no terminal.

5.2. Executar o comando para gerar plots

- Execute o comando:

```
java -jar GenerateDataPlot.jar
```

5.3. Executar os scripts de bash

- No terminal, vá até a pasta `RunExperiments/results`;
- Execute os seguintes scripts para gerar os gráficos:

```
./launchLength.sh
./launchQuality.sh
./launchRuntime.sh
```

5.4. Verificar os resultados

- Os resultados dos gráficos estarão disponíveis na pasta `RunExperiments/results/NbIterations/` com o nome do dataset correspondente.
- Os gráficos gerados estarão em formato `.pdf` e prontos para análise.

6. Executar Experimentos em Lotes

A pasta `RunExperiments` contém o programa que pode executar automaticamente o algoritmo para todos os datasets. Nesta pasta, a pasta `results` contém os resultados dos lotes, a pasta `src` contém os códigos-fonte deste programa, o arquivo `RunExperiments.jar`, e o arquivo `paramGen.conf` para configuração, assim como é o arquivo `parameters.conf`. Para executar, use o comando:

```
java -jar RunExperiments.jar [options]
```

As opções disponíveis são escolhidas entre {ucb, expand, rollout, memory, update, iter}, que permitem executar o experimento de uma estratégia definida de MCTS.

- **ucb**: Executa experimentos usando a estratégia Upper Confidence Bound (UCB).
- **expand**: Executa experimentos focando na expansão da árvore.
- **rollout**: Executa experimentos utilizando a estratégia de rollout.
- **memory**: Executa experimentos com foco em otimizações de memória.
- **update**: Executa experimentos aplicando estratégias de atualização.
- **iter**: Executa experimentos através de iterações específicas.

Para iniciar o programa com uma opção específica, use o comando no terminal, por exemplo:

```
java -jar RunExperiments.jar ucb
```

Os resultados dos experimentos serão armazenados na pasta `results` dentro de uma hierarquia definida, facilitando a análise dos diferentes experimentos realizados.