



PUC-Rio - Pontifical Universidade Católica do Rio de Janeiro

SAD - Sistema de Avaliação de Desempenho
Projeto de Software
Projeto de Programação 2014.1

SAD - Sistema de Avaliação de Desempenho

Projeto de Software

Versão 1



PUC-Rio - Pontifícia Universidade Católica do Rio de Janeiro

SAD - Sistema de Avaliação de Desempenho
Projeto de Software
Projeto de Programação 2014.1

Histórico de Revisões

Versão	Data	Autor	Conteúdo
1	25/04/2014	Daniel Marques	Primeira versão do documento.
2	25/07/2014	Daniel Marques	Revisão do documento.

Documentos de Referência

Título	Versão	Data	Autor
Manual do Weka	3.7.10	31/07/2013	Bouckaert, et. al.
Especificação de Requisitos	1	14/04/2014	Daniel Marques
Programing in Lua	3th ed.	2013	Ierusalimschy, Roberto

Introdução

Este é o documento de projeto de software do SAD (Sistema de Avaliação de Desempenho). Software a ser desenvolvido na disciplina Projeto de Programação da PUC-Rio. O projeto do sistema baseia-se no que foi definido na especificação de requisitos do sistema SAD. Este documento deve ser usado como referência a codificação do sistema.

Escopo

Conforma definido na especificação de requisitos, o SAD é um em um sistema de avaliação de desempenho de métodos. Seu objetivo é dar suporte as técnicas de aprendizado de máquina.

O sistema SAD deve receber como entrada um conjunto de dados e um conjunto de métodos simples. Ele aplicará os métodos aos dados e gerará estatísticas de acerto. Estas estatísticas podem revelar a pertinência da aplicação de técnicas de aprendizado de máquina. Se for o caso, estas também podem ser usadas como valores de referência para os resultados do aprendizado de máquina.



Figura 1 - Diagrama simplificado do sistema SAD

Definições, Acrônimos e Abreviaturas

Weka - Colação de algoritmos de aprendizado de máquina para mineração de dados.

Lua - Linguagem de programação.

Tecnologias

O sistema ser3 desenvolvido na linguagem de programaa33o Lua. O sistema 3 portanto compatfvel com qualquer plataforma que suporte a linguagem lua. O sistema de versionamento de c3digo fonte Git foi utilizado para versionar o sistema. O c3digo fonte do SAD pode ser obtido no link: <https://github.com/danielmarques/sad.git>.

Decomposi33o em M3dulos

Diagrama de Pacotes

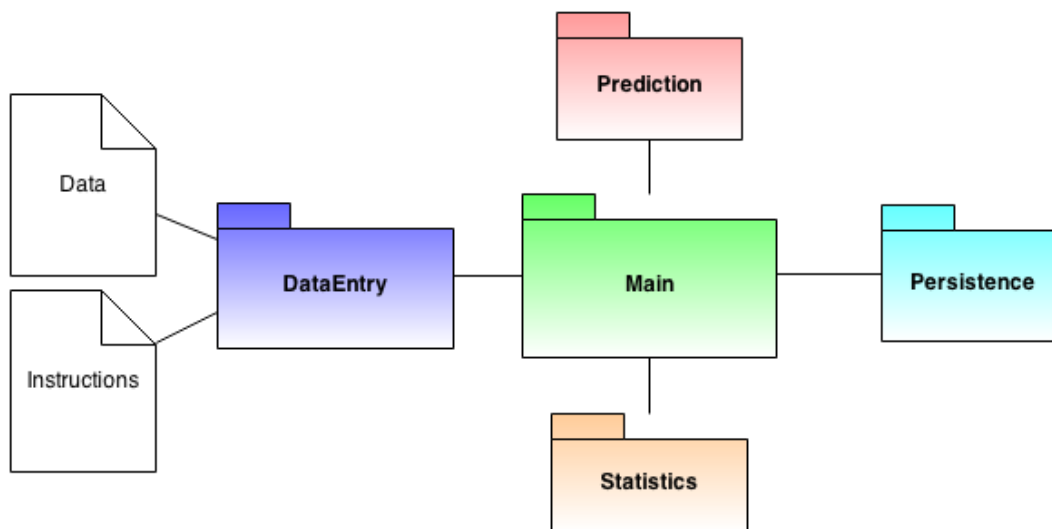


Figura 2 - Diagrama de Pacotes

Diagrama de Componentes

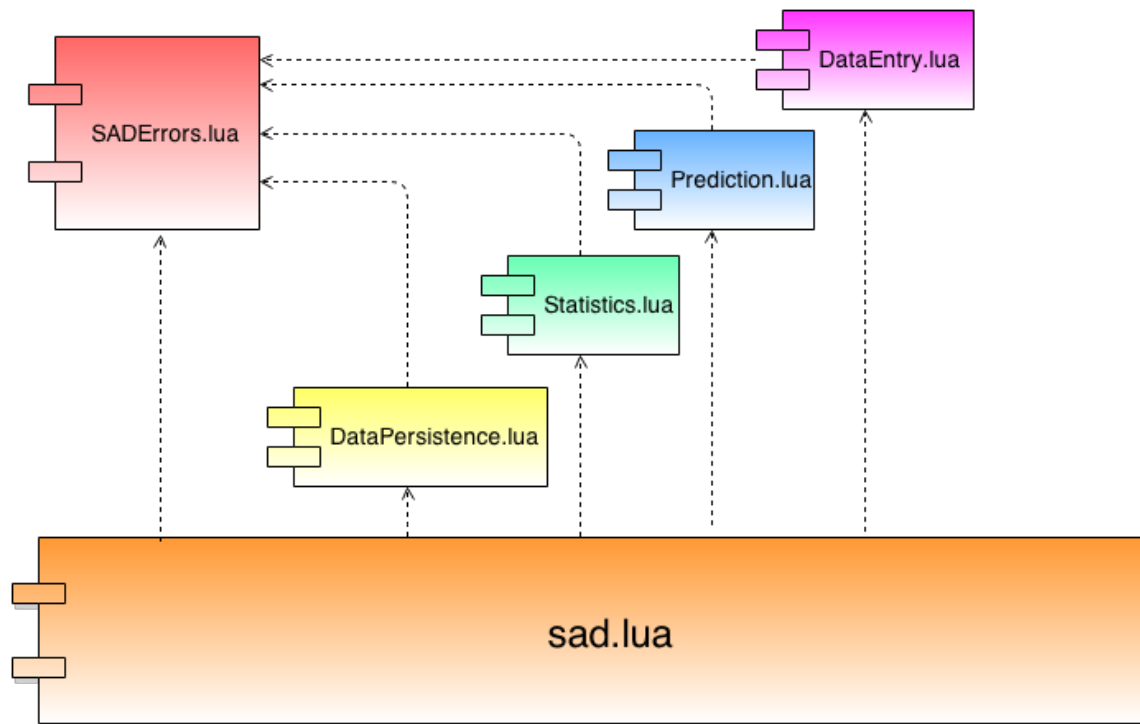


Figura 3 - Diagrama de Componentes

Descrição dos Módulos

Descrição detalhada dos módulos, suas dependências, interfaces e entidades de dados. A modularização do sistema foi feita de acordo com as políticas estabelecidas no capítulo 15 do livro *Programming in Lua: Modules and Packages*.

Módulo Principal

Módulo que interpreta e executa os experimentos definidos no arquivo de instruções. O módulo principal utiliza as funcionalidades implementadas nos outros módulos para executar as tarefas. Este módulo executa chamadas aos outros módulos em modo protegido a fim de tratar erros. Ele também é responsável por formatar e apresentar os resultados ao usuário.

Componentes: `sad.lua`

Módulo de Entrada de Dados

Módulo responsável por carregar o conteúdo dos arquivos de instrução e dados. Este módulo disponibiliza as informações dos arquivos por meio de tabelas lua.

A tabela de instruções é um *array*, ou seja uma tabela com chaves numéricas sequenciais, onde cada elemento também é uma tabela lua com o formato:

```
{  
    data = "<Caminho para o arquivo de dados 2>",  
    strategy = "<Nome da estratégia de predição 2>",  
    strategy option = "<valor da opção da estratégia>", <opção 2> = <valor opção 2>, ..."  
}
```

Note que o *strategy option* pode ter um identificador diferente para cada estratégia.

A tabela de dados é uma *array* onde cada elemento é uma tabela lua. Esta última tabela representa uma instância dos dados. O módulo de importação de dados acrescenta um único valor a cada instância, identificado pela chave *real*. Este representa o valor real da instancias, retirado o arquivo de dados e pode assumir um valor *numeric* ou *string*.

Componentes do módulo: DataEntry.lua

Módulo de Predição

Módulo que disponibiliza múltiplas estratégias de predição que podem ser aplicadas aos dados. cada estratégia é implementada por uma função. As estratégias de predição são:

- Repetir o anterior (*RepeatPrevious*)
- Média de todos (*MeanAll*)
- Média dos p últimos (*MeanP*)
- Mais numeroso (*MostNumerousAll*)
- Mais numeroso entre os p últimos (*MostNumerousP*)

Todas as estratégias devem receber como entrada uma tabela lua (conforma a definida no módulo de entrada de dados). Depois de executar a predição a chave *predicted* é acrescentada a cada instância dos dados, seu valor pode ser *numeric* ou *string*.

Componentes do módulo: Prediction.lua



Módulo de Estatísticas

Módulo que disponibiliza funções que comparam o resultado da predição com o resultado real para então gerar as estatísticas de resultado. As funções disponíveis são:

- *Correctness*: Função que efetua a contagem das instâncias corretamente e incorretamente classificada
- *MeanAbsError*: Função que calcula o erro médio absoluto.
- *RootMeanSqrdError*: Função que calcula o erro quadrático médio

Componentes do módulo: Statistics.lua

Módulo de Persistência

Módulo que disponibiliza funções que permitem salvar os resultados em arquivos. As funções disponíveis são:

- *SaveStatistics*: Salva uma string em arquivo.
- *SaveData*: Sava dados (conforme definido no módulo Predição) em arquivo.

Componentes do módulo: Persistence.lua

Módulo de Erros

Módulo que define as mensagens de erro do sistema.

Componentes do módulo: SADErros.lua