DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
Instituto Superior de Engenharia do Porto

Trabalho de LAPR1
Análise de Séries Temporais

Dezembro 2019

1 Objetivos

Com este projeto pretende-se que os alunos desenvolvam uma aplicação em linguagem Java onde apliquem um processo básico de desenvolvimento de aplicações informáticas, valorizando todas as fases do ciclo de desenvolvimento, desde a análise e conceção aos testes de validação. Pretende-se também que os alunos elaborem um relatório que descreva a aplicação concebida, o processo de desenvolvimento e que apresentem e critiquem os resultados obtidos.

Em particular, no projeto a realizar no corrente ano letivo pretende-se que os alunos estudem métodos e técnicas que permitam analisar Séries Temporais (Hyndman & Athanasopoulos, 2014; SEMATECH, 2019) e estimar valores futuros (fazer uma previsão).

2 Análise de Séries Temporais

Uma série temporal é uma sequência de observações ordenada cronologicamente que, em geral, são recolhidos em intervalos regulares. A análise de séries temporais pode ser aplicada a qualquer variável que muda ao longo do tempo e, de um modo geral, as observações mais próximas têm valores mais próximos e correlacionados que aqueles valores mais distantes (Hyndman & Athanasopoulos, 2014; SEMATECH, 2019).

A análise de séries temporais é de grande utilidade em vários domínios como as Finanças, Meteorologia, Energia, entre outros, sendo que do seu processamento podem resultar ganhos significativos para o conhecimento do negócio ou planeamento de atividades.

Para extrair conhecimento das séries temporais existe um conjunto alargado de métodos, técnicas e ferramentas que podem ser utilizados. Alguns destes métodos exigem conhecimentos avançados que são apenas adquiridos através de formação avançada ao nível de formação superior especializada. Outros métodos, como os que serão explorados neste projeto, são mais simples e podem ser estudados e implementados por qualquer aluno que frequenta um curso superior de engenharia ou ciências. Entre estes estão métodos e técnicas básicos que permitem analisar uma série temporal em diferentes resoluções, filtrar/suavizar uma série e prever valores futuros utilizando modelos.

2.1 Análise utilizando diferentes resoluções

As observações que constituem um série temporal podem ser recolhidas com elevada frequência, para permitir monitorizar e analisar com rigor ocorrências, ou com menor frequência, quando o estado de um sistema ou processo praticamente não é alterado ao longo do tempo e uma amostra recolhida com grande espaçamento é suficiente para analisar a evolução de um sistema ou processo. Em geral, e atendendo à tecnologia disponível, recolhem-se dados com elevada frequência que depois são transformados para a resolução necessária ao objetivo e problema que se pretende resolver. Por exemplo, se pretendemos analisar o consumo de eletricidade mensal de uma casa e estamos a recolher dados que representam o consumo a cada hora, então teremos que somar o consumo de todos os dias e horas de um determinado mês para calcular o consumo total nesse mês.

2.2 Filtragem/Suavização

Uma técnica relevante para analisar uma série temporal é a filtragem. Ao filtrar os dados podemos remover ruído e identificar tendências. Entre as técnicas de filtragem mais utilizadas e simples estão a Média Móvel Simples e a Média Móvel Exponencialmente Pesada (SEMATECH, 2019).

A **Média Móvel Simples** é definida através da equação:

$$y_i = \frac{1}{n}\sum_{k=0}^{n-1} x_{i-k},$$

onde x_i são os termos que representam a série original, y_i é a série resultante da aplicação do filtro (da suavização) e n é a ordem da média móvel.

A **Média Móvel Exponencialmente Pesada** é definida através da equação:

$$y_i = \alpha x_i + (1-\alpha)y_{i-1},$$

onde x_i são os termos que representam a série original, y_i é a série resultante da aplicação do filtro (da suavização) e α é uma constante que toma valores no intervalo $]0,1]$. Na Figura 1 é apresentada uma série original e duas séries suavizadas que resultam da aplicação do modelo Média Móvel Exponencialmente Pesada, para dois valores de α distintos (0.3 e 0.05).

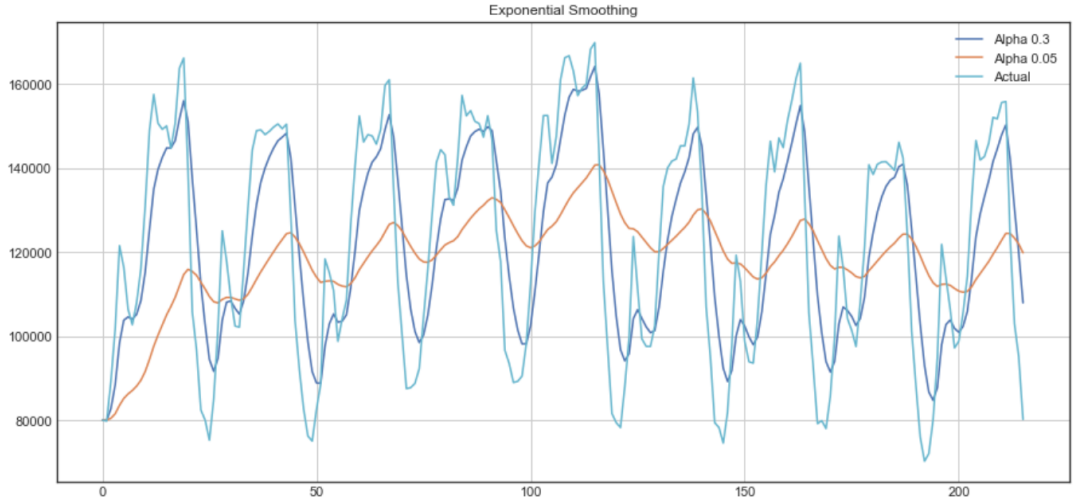

Figura 1: Exemplo de aplicação de Média Móvel Exponencialmente Pesada. Imagem retirada de (Peixeiro, 2019)

2.3 Previsão

Para analisar uma série temporal também é possível encontrar um modelo matemático que capture o processo que gerou a série temporal e permita prever valores futuros da série utilizando o histórico de dados. Entre os modelos mais utilizados e simples estão os modelos de Média Móvel Simples e os modelos de Média Móvel Exponencialmente Pesada apesentados na Secção 2.2.

Para realizar uma previsão utilizando a **Média Móvel Simples** recorremos à equação:

$$y_{i+1} = \frac{1}{n}\sum_{k=0}^{n-1} x_{i-k},$$

onde x_i são os termos que representam a série original, y_{i+1} representa uma previsão utilizando dados históricos e n é a ordem da média móvel.

No caso da **Média Móvel Exponencialmente Pesada**, para realizar uma previsão recorremos à equação:

$$y_{i+1} = \alpha x_i + (1 - \alpha) y_i,$$

onde x_i são os termos que representam a série original, y_{i+1} representa uma previsão utilizando dados históricos e α é uma constante que toma valores no intervalo $]0, 1]$.

3 Trabalho a Desenvolver

O trabalho desenvolvido neste projeto será entregue em duas fases, primeiro a aplicação e mais tarde o relatório juntamente com uma apresentação em PowerPoint.

Assim, o trabalho a realizar até ao dia **22 de Dezembro de 2019** consiste em:

- Estudar Análise de Séries Temporais, em particular os métodos e técnicas básicos que permitem analisar uma série temporal em diferentes resoluções, modelar uma série e prever valores futuros baseados em modelos que foram apresentados na Secção 2.
- Desenvolver uma aplicação que permita:
  - Analisar series temporais em diferentes resoluções temporais considerando: periodos do dia (Manhã: 6:00 às 11:59; Tarde: 12:00 às 17:59; Noite: 18:00 às 23:59; Madrugada: 00:00 às 05:59), diário, mensal, anual.
  - Calcular o número de observações que ocorrem num conjunto de intervalos. Neste trabalho devem considerar três intervalos que são definidos utilizando a média global da série (μ): $[-\infty, \mu - 0.2\mu[$, $[\mu - 0.2\mu, \mu + 0.2\mu[$ e $[\mu + 0.2\mu, \mu + \infty[$.
  - Ordenar os valores da série temporal por ordem crescente ou decrescente, conforme a opção do utilizador. O algoritmo a utilizar deve ser o mais eficiente possível. Para escolher este algoritmo os alunos devem implementar e avaliar o desempenho de três algoritmos de ordenação, *Insertion Sort*, *Bubble Sort* e *Merge Sort* (Horstmann, 2015), e escolher o melhor com base no tempo necessário para ordenar o ficheiro de entrada disponibilizado no moodle. Este processo de seleção do melhor algoritmo deve ser descrito no relatório.

    Nota: Este método de escolha do melhor algoritmo de ordenação a utilizar carece de rigor e só é utilizado porque os alunos ainda não adquiriram os conhecimentos necessários para fazer uma seleção rigorosa. Ainda assim, os alunos podem considerar um maior número de conjuntos de dados e com diferente dimensão.
  - Implementar a Média Móvel Simples e Média Móvel Exponencialmente Pesada (conforme Secção 2.2) para filtrar a série temporal original. A aplicação deve permitir que o utilizador defina cada um dos parâmetros necessários (n e α).
  - Calcular o Erro Médio Absoluto (MAE) que se comete ao aproximar a série observada com um dos modelos de filtragem da Secção 2.2. O erro médio absoluto é dado pela fórmula: $MAE = \frac{\sum_{i=0}^{N-1} |y_i - x_i|}{N}$, em que x_i são os termos que representam a série original, y_i é a série resultante da aplicação do filtro e N é o número de observações da série.
  - Prever o valor de uma observação futura utilizando um dos modelos de previsão (ver Secção 2.3) escolhidos pelo utilizador. O utilizador especifica um periodo do dia, um dia

ou um mês, conforme a série a ser analisada, e a aplicação apresenta a previsão para esse mesmo momento.
  - Permita carregar séries temporais armazenadas em ficheiros CSV (*Comma-separated values*), sendo que o nome do ficheiro descreve a origem da série temporal e o conteúdo descreve o momento da observação e o valor observado.
  - Visualizar toda a informação na consola excepto os gráficos que devem ser visualizados utilizando o *gnuplot*. Sempre que um gráfico é gerado deve ser dada a opção ao utilizador para gravar o gráfico em formato PNG e para gravar a série temporal em ficheiro CSV.

- A aplicação deve ter uma interface simples e intuitiva que permita selecionar qualquer das funcionalidades e visualizar o respetivo resultado. Sempre que possível os resultados devem ser apresentados através de um gráfico. Para este fim deve ser utilizada a biblioteca *JavaPlot* (http://javaplot.panayotis.com/index.html) que consiste num interface para a aplicação *gnuplot* (Williams & Kelley, 2010).

O trabalho a realizar até ao dia **6 de Janeiro de 2020** consiste em:

- Elaborar um relatório em que são descritos: os métodos e técnicas estudados; a metodologia de trabalho que utilizaram para desenvolver a aplicação; a implementação da aplicação; e a análise de resultados. A descrição da implementação da aplicação deve incluir um diagrama que identifique claramente os módulos e suas dependências. A apresentação dos métodos e técnicas deve incluir exemplos ilustrativos.

- Preparar uma apresentação PowerPoint com não mais que 8 slides. Esta apresentação será utilizada para fazer uma apresentação do trabalho, com a duração de dez minutos, no momento de avaliação do projeto.

4 Formato dos Dados de Entrada e Saída

Os dados de entrada para a aplicação são ficheiros que representam o consumo horário de energia, em megawatts (MW), em regiões dos Estados Unidos da América[1]. Na UC de LAPR1 exploramos estes dados restringindo a subconjuntos de dados que representam no máximo três anos de leituras e em que não há falhas, a cada hora é registado o consumo de uma região. Cada um dos ficheiros tem um cabeçalho e todos os campos dos ficheiros estão separados por uma vírgula. O nome do ficheiro identifica a região onde foi recolhida a série temporal.

Um exemplo de um ficheiro que pode ser utilizado como entrada na aplicação a desenvolver está disponível no moodle da unidade curricular.

O formato dos dados de saída está dependente da operação realizada e devem apresentar de forma clara a informação. Todos os resultados devem ser apresentados na consola excepto os gráficos, que devem ser apresentos em imagens. Dependendo do interesse do utilizador, as séries temporais resultantes da transformação também podem ser gravadas em ficheiro PNG e/ou CSV, sendo que estes ficheiros devem ter um nome que permita identificar a serie temporal que foi utilizada como entrada, o tipo de agregação, o tipo de filtragem e o momento em que o ficheiro foi gerado.

5 Método de Trabalho

- Todos os alunos devem utilizar a metodologia de trabalho definida no *eduScrum* (Delhij & Solingen, 2013). Cada um dos grupos deve escolher um Scrum Master e este deve ser responsável

[1] https://www.kaggle.com/robikscube/hourly-energy-consumption

por gerir a execução de tarefas. Para atingir este objetivo, o grupo deve utilizar a ferramenta *Trello*[2] e registar as tarefas do projeto, a atribuição de tarefas, o estado de cada tarefa e as tarefas concluídas.

- A aplicação será desenvolvida utilizando o sistema de controle de versões *Git* e o *Bitbucket*[3]. Todos os alunos terão que criar uma conta no *Bitbucket* com o endereço de email do ISEP (i.e. 1XXXXXX@isep.ipp.pt) e cada grupo terá que criar um repositório. A designação do repositório deve seguir o formato do exemplo "LAPR1_TurmaDAB_Grupo01". O repositório deve ser partilhado com todos os docentes que lecionam a turma onde o grupo está inserido.

- O grupo deve criar uma pasta no *OneDrive*[4] onde guarda todo o material desenvolvido para a realização do projeto. A designação da pasta deve seguir o formato do exemplo "LAPR1_ TurmaDAB_Grupo01". A pasta será partilhada com todos os docentes que lecionam a turma onde o grupo está inserido. Não é necessário incluir nesta pasta o código que está disponível no repositório do *BitBucket*.

6 Processo de Desenvolvimento de Software

- A aplicação deve ser estruturada e organizada em módulos. Será valorizada uma correta decomposição modular e o reaproveitamento de módulos.

- O trabalho deverá ser desenvolvido em linguagem Java e deverá resultar num ÚNICO projeto NetBeans.

- A visualização de gráficos será implementada utilizando a biblioteca JavaPlot[5].

- A aplicação pode ser executada em modo interativo ou não interativo.

  - No modo interativo a aplicação deverá ser chamada da linha de comandos utilizando o comando: *java -jar nome_programa.jar -nome ts_nome da serie temporal.csv*. Neste modo interativo todos os parâmetros necessários serão solicitados em tempo de execução. Todos os resultados, excepto os gráficos, devem ser apresentados na consola. Relativamente à apresentação de uma série temporal, esta deve ser feita utilizando gráficos mas dando sempre a possibilidade ao utilizador de gravar a série temporal em ficheiros PNG e/ou CSV.
  - No modo não interativo não há qualquer interação com o utilizador e pretende-se que sejam executadas todas as funcionalidades da aplicação através de um único comando, em que são especificados todos os parâmetros. Para executar a aplicação neste modo, esta deve ser chamada da linha de comandos utilizando o comando: *java -jar nome_programa.jar -nome ts_nome_da_serie_temporal.csv -resolucao X -modelo M -tipoOrdenacao T -parModelo nAlpha -momentoPrevisao D*, em que X, M, T, nAlpha e D são parâmetros que podem tomar os seguintes valores: X pode tomar os valores 11, 12, 13, 14, 2, 3 e 4 que representam respetivamente os periodos manhã, tarde, noite, madrugada, diário, mensal e anual; M pode tomar os valores 1 e 2 que representam respetivamente os modelos Média Móvel Simples e Média Móvel Exponencialmente Pesada; T pode tomar valores 1 ou 2, sendo que 1

[2] https://trello.com/
[3] https://bitbucket.org
[4] https://onedrive.live.com
[5] http://javaplot.panayotis.com/index.html

representa ordenação crescente e 2 ordenação decrescente; nAlpha toma um valor numérico que corresponde ao parâmetro n ou parâmetro *alpha*, conforme o valor do parâmetro M; D representa um momento de previsão e o seu valor depende da resolução específicada. Neste modo todo o output deve ser guardado num ficheiro de texto e, no caso dos gráficos, também em ficheiros PNG. As séries temporais que são apresentadas em gráficos, e guardadas em ficheiros PNG, também devem ser guardadas em ficheiros CSV.

- Todos os métodos desenvolvidos terão, obrigatoriamente, de estar associados a testes unitários. Por exemplo, se o aluno criar o método $ordenar\_crescente\_serie(x)$ para ordenar por ordem crescente uma série x, também deve criar o método $test\_ordenar\_crescente\_serie(x, expected\_x\_ordenada)$ (ver Algoritmo 1), em que *expected_x_ordenada* é uma ordenação prévia da série e que serve para verificar se o algoritmo de ordenação teve o comportamento esperado. Estes testes são extremamente úteis para determinar se os métodos estão de acordo com a sua especificação e se a edição destes não alterou a funcionalidade.

```
Bool test_ordenar_crescente_serie(x, expected_x_ordenada)
{

x_ordenada = ordenar_crescente_serie(x);

if(expected_x_ordenada==x_ordenada)
   return True;
else
   return False;

}
```

Algoritmo 1: Exemplo de teste unitário

7 Submissão do Trabalho

Datas e entregas de trabalho a efetuar através do Moodle:

- Dia 22 de Dezembro de 2019, até às 23h00m
  - Submeter o projeto desenvolvido, **versão final**, incluindo toda a estrutura de diretorias e ficheiros do projeto (incluindo o executável), num único ficheiro comprimido (ZIP).
- Dia 6 de Janeiro de 2020, até às 23h00m
  - Relatório em formato pdf não ultrapassando as 25 páginas. A escrita do relatório deve seguir as instruções formais e o modelo disponibilizado nas aulas TP (módulo de competências).
  - Apresentação em PowerPoint com não mais que 8 slides. A apresentação deve seguir as instruções formais e o modelo disponibilizado nas aulas TP (módulo de competências).

Nota: Os ficheiros deverão identificar, obrigatoriamente, a designação do grupo e a turma a que os alunos pertencem (Exemplo: "LAPR1_TurmaDAB_Grupo01_projeto.ZIP"; "LAPR1_TurmaDAB_Grupo01_relatorio.PDF"e "LAPR1_TurmaDAB_Grupo01_relatorio.PPT").

Referências

Delhij, A., & Solingen, R. (2013). *The eduscrum guide: The rules of the game.* (Disponível em http://eduscrum.nl/file/CKFiles/The_eduScrum_Guide_EN_December_2013_1.0.pdf)

Horstmann, C. (2015). *Big java: Early objects, 6th edition.* Wiley. Retrieved from `https://books.google.pt/books?id=ib12CwAAQBAJ`

Hyndman, R., & Athanasopoulos, G. (2014). *Forecasting: principles and practice.* OTexts. Retrieved from `https://books.google.pt/books?id=gDuRBAAAQBAJ`

Peixeiro, M. (2019). *Almost everything you need to know about time series.* (Disponível em https://towardsdatascience.com/almost-everything-you-need-to-know-about-time-series-860241bdc578)

SEMATECH. (2019). *Engineering statistics handbook.* National Institute of Standards and Technology. (Disponível em https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm)

Williams, T., & Kelley, C. (2010, March). *gnuplot 5.0: An interactive plotting program.* `http://gnuplot.sourceforge.net/`.