ME524: Computação Aplicada à Estatística, Atividade II

Guilherme Ludwig
`gvludwig@ime.unicamp.br`

14 de novembro de 2024

- A atividade é individual. Referencie o trabalho citado, mas desenvolva as soluções na atividade. Explique resultados com suas próprias palavras.
- A solução deve ser desenvolvida em detalhe. Todas as contas devem estar explicadas e desenvolvidas.
- Cada aluno deverá ser responsável por garantir que não há cópia do seu trabalho. Indícios de plágio podem levar à anulação da atividade de todos os envolvidos. Use seu bom senso para guiar as conversas com seus colegas.
- A atividade deve ser entregue através do Moodle, em um único documento em formato `PDF`. No máximo 6 páginas de teoria, discussão, resultados etc.; um apêndice deve conter todo o código sequencial. Em particular, de tal forma que eu possa copiar e colar no meu computador, e obter exatamente os mesmos resultados do seu relatório. Se o apêndice for mais longo que duas páginas, você provavelmente está programando mal. Os códigos vão ser analisados para eficiência, também.
- Use a linguagem que você preferir, de preferência com o pacote `listings` do LaTeX para exibir código: `R`, `C`, `python`, `julia`, ou outra qualquer. Todos os códigos devem estar listados no final da atividade, com instruções de execução (à parte das linguages listadas).
- Se você quiser, pode usar `Rmarkdown`, `Jupyter notebooks` ou o que preferir para mesclar código e texto.
- **Importante:** Usar pacotes que já implementam as regressões resultará em nota zero. Usar pacotes (como o `numDeriv`) para calcular as derivadas resultará em grande penalização da nota; mesma coisa para o `optim` (que vocês podem usar para verificar resultados, mas não para resolver as questões). Desenvolvam teoria e apresentem código. O computador neste curso é o instrumento, não deve "pensar" no seu lugar.
  - De preferência, não evoquem **nenhum** pacote. No máximo algo para visualização (embora, pessoalmente, eu ache a visualização do `base` superior nesses problemas não-lineares). Uma sessão limpa do R tem tudo o que você precisa.
- **Prazo de entrega:** 26/11/2024, às 23:59, via Moodle. Note que já incluí extensão no prazo original, solicitada pelos alunos. Sem tolerância para atraso.

Este projeto está adaptado de um exercício em Greene (2008, p.240). O conjunto de "dados" a seguir é, na verdade, um conjunto de modelos que servem para testar a precisão de programas de computadores, feito pelo *National Institute of Standards and Technology* (NIST). Vamos considerar dois modelos de regressão não-linear, que encontram-se na URL `http://www.itl.nist.gov/div898/strd/`, para verificar a implementação feita por vocês dos métodos de Newton-Raphson, Gradient Descent e Line Search.

- O primeiro modelo é `Misra1a` (`https://www.itl.nist.gov/div898/strd/nls/data/misra1a.shtml`), um modelo não-linear para estudos de adsorção de fluidos nos dentes (cf. referência), dado por

  $$y_i = f_1(x_i, \boldsymbol{\beta}) + \varepsilon_i = \beta_1(1 - \exp\{-\beta_2 x_i\}) + \varepsilon_i, \quad i = 1, \ldots, n.$$

  Aqui os parâmetros β_1, β_2 são reais e desconhecidos. O conjunto de dados correspondente pode ser declarado no R com o comando:

```
Misra1a <- data.frame(y = c(10.07, 14.73, 17.94, 23.93, 29.61,
                            35.18, 40.02, 44.82, 50.76, 55.05,
                            61.01, 66.4, 75.47, 81.78),
                      x = c(77.6, 114.9, 141.1, 190.8, 239.9,
                            289, 332.8, 378.4, 434.8, 477.3,
                            536.8, 593.1, 689.1, 760))
```

Compare a performance dos algoritmos com dois valores iniciais $\hat{\boldsymbol{\beta}}^{(0)} = (500, 0.0001)^T$, e $\hat{\boldsymbol{\beta}}^{(0)} = (250, 0.0005)^T$.

- O segundo modelo é `Thurber` (`https://www.itl.nist.gov/div898/strd/nls/data/thurber.shtml`), um modelo não-linear para mobilidade de elétrons em semicondutores, dado por

$$y_i = f_2(x_i, \boldsymbol{\beta}) + \varepsilon_i = \frac{\beta_1 + \beta_2 x_i + \beta_3 x_i^2 + \beta_4 x_i^3}{1 + \beta_5 x_i + \beta_6 x_i^2 + \beta_7 x_i^3} + \varepsilon_i, \quad i = 1, \ldots, n.$$

Aqui os parâmetros $\beta_1, \beta_2, \ldots, \beta_7$ são reais e desconhecidos. O conjunto de dados correspondente pode ser declarado no R com o comando:

```
Thurber <- data.frame(y = c(80.574, 84.248, 87.264, 87.195, 89.076, 89.608,
                            89.868, 90.101, 92.405, 95.854, 100.696, 101.06,
                            401.672, 390.724, 567.534, 635.316, 733.054,
                            759.087, 894.206, 990.785, 1090.109, 1080.914,
                            1122.643, 1178.351, 1260.531, 1273.514, 1288.339,
                            1327.543, 1353.863, 1414.509, 1425.208, 1421.384,
                            1442.962, 1464.35, 1468.705, 1447.894, 1457.628),
                      x = c(-3.067, -2.981, -2.921, -2.912, -2.84, -2.797,
                            -2.702, -2.699, -2.633, -2.481, -2.363, -2.322,
                            -1.501, -1.46, -1.274, -1.212, -1.1, -1.046,
                            -0.915, -0.714, -0.566, -0.545, -0.4, -0.309,
                            -0.109, -0.103, 0.01, 0.119, 0.377, 0.79, 0.963,
                            1.006, 1.115, 1.572, 1.841, 2.047, 2.2))
```

Compare a performance dos algoritmos com dois valores iniciais $\hat{\boldsymbol{\beta}}^{(0)} = (1000, 1000, 400, 40, 0.7, 0.3, 0.03)^T$, e $\hat{\boldsymbol{\beta}}^{(0)} = (1300, 1500, 500, 75, 1, 0.4, 0.05)^T$.

Para resolver a atividade, você deve seguir o seguinte roteiro:

- Considerando a perda quadrática $Q_j(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i - f_j(x_i, \boldsymbol{\beta})\right]^2$, para os problemas $j = 1, 2$, encontre os respectivos gradientes e matrizes Hessianas, **explicitamente**. Implemente o procedimento de **minimização** das funções Q_j, usando o método de Newton-Raphson, nos pontos iniciais indicados.
- Investigue a possibilidade de *line-search* com tamanho γ para o método de Newton-Raphson. Neste cenário, se as derivadas não forem razoáveis, considere o método de bissecção para $\gamma \in (0, 1)$. Justifique.
- Implemente o procedimento de **minimização** das funções Q_j, usando o método de *gradient descent*, nos pontos iniciais indicados. Investigue a possibilidade de *line-search* com tamanho γ.
- Nos dois casos, a sua função de otimização deve ser implementada manualmente, para uma tolerância arbitrária (escolha e justifique). Compare as estimativas obtidas com os coeficientes de referência estimados nas respectivas páginas. Nos dois casos, conte o número de operações que você executou, até atingir a tolerância. Você pode fazer figuras com as funções indicadas.

Referências

W. H. Greene. *Econometric Analysis*. Pearson, New York, NY, 2008.