

MÉTODOS COMPUTACIONALES AVANZADOS

Jaime E. Forero Romero

NOMBRE DEL CURSO: Métodos Computacionales Avanzados

CÓDIGO DEL CURSO: FISI 3028 / FISI 4028

UNIDAD ACADÉMICA: Departamento de Física

PERIODO ACADÉMICO: 201820

HORARIO: Mi 6:30 a 7:50 y Vi 6:30 a 7:50

NOMBRE PROFESOR(A) PRINCIPAL: Jaime E. Forero Romero

CORREO ELECTRÓNICO: je.forero@uniandes.edu.co

NOMBRE PROFESOR(A) COMPLEMENTARIO(A):

CORREO ELECTRÓNICO: @uniandes.edu.co

I Introducción

Los métodos computacionales fundamental el trabajo en todas las áreas técnicas y científicas, ya sean principalmente experimentales o teóricas. Esto se debe en gran parte a que la capacidad de utilizar computadoras de alto rendimiento ha disminuido en costo monetario y en complejidad.

El curso de Métodos Computacionales Avanzados presenta estas posibilidades computacionales a estudiantes de diferentes disciplinas científicas. Para esto se propone profundizar sus conocimientos en dos áreas: implementación de métodos de aprendizaje estadístico (i.e. Machine Learning, algoritmos que aprenden de datos) y utilización de técnicas de cómputo masivamente paralelo.

Se asume que los estudiantes de este curso ya tienen conocimientos básicos en métodos computacionales equivalentes al nivel del curso Métodos Computacionales (FISI-2028).

II Objetivos

Los objetivos principales del curso son:

- Demostrar aplicaciones de métodos de estadística bayesiana para el análisis estadístico de datos.
- Demostrar el uso de métodos y librerías de aprendizaje estadístico (Machine Learning).
- Mostrar diferentes arquitecturas, paradigmas, lenguajes y librerías para cómputo masivamente paralelo.
- Estudiar diferentes aplicaciones prácticas a problemas científicos y del contexto industrial de las metodologías computacionales modernas.

III Competencias a desarrollar

Al finalizar el curso, se espera que el estudiante esté en capacidad de:

- Manejar lenguajes modernos de computación numérica de bajo nivel (i.e. C/C++) y de alto nivel (i.e. Python/R/Julia).
- Tener un esquema para pre-procesar, analizar y generar reportes científicos y técnicos a partir de diversas fuentes de datos utilizando métodos computacionales.
- Desplegar programas en paralelo en un cluster
- Generar conocimiento a partir del modelamiento teórico y computacional de los conceptos vistos en clase.

IV Contenido por semanas

Semana 1. Binder. Jupyterlab. Fundamentos de python, numpy y matplotlib.

Semana 2. Fundamentos de probabilidad y Teorema de Bayes.

Referencia: Capítulos 1 de DABT

Semana 3. Estimación de parámetros con estadística bayesiana. Algoritmo de Metropolis-Hastings.
Referencia: Capítulo 29 de ITILA. Capítulos 2 y 3 de DABT.

Semana 4. Algoritmos Monte Carlo eficientes: Nested Sampling, Hamiltoniano
Referencia: Capítulo 30 de ITILA. Capítulo 9 de DABT.

Semana 5. Selección de Modelos
Referencia: Capítulo 4 de DABT.

Semana 6. Introducción a Machine Learning.
Referencia: Capítulos 1 y 2 de ISLR.

Semana 7. Regresión lineal y cross-validation
Referencia: Capítulos 3 y 5 de ISLR.

Semana 8. Cross-validation y regularización.
Referencia: Capítulos 5 y 6 de ISLR.

Semana 9. Semana de receso.

Semana 10. Clasificación.
Referencia: Capítulo 4 ISLR.

Semana 11. Árboles de Decisión.
Referencia: Capítulo 8 ISLR.

Semana 12. Support Vector Machines.
Referencia: Capítulo 9 ISLR.

Semana 13. PCA, Clustering, K-means. t-SNE.
Referencia: Capítulo 10 ISLR.

Semana 14. Fundamentos de C. Makefiles. Taxonomía de arquitecturas para cómputo en paralelo. Máquinas para cómputo en paralelo en Uniandes.
Referencia: Capítulo 11 del libro de ISTC.

Semana 15. Fundamentos de programación en paralelo: MPI.
Referencia: Capítulo 12 del libro de ISTC.

Semana 16. Fundamentos de programación en paralelo: OPENMP.
Referencia: Capítulo 13 del libro de ISTC.

V Metodología

Cada semana tendremos una corta presentación teórica (media hora aproximadamente) para pasar a ejercitar esos conceptos directamente en la computadora/cluster haciendo ejercicios de práctica (dos horas aproximadamente). Los estudiantes deben leer la bibliografía recomendada **antes** de las clases correspondientes.

VI Criterios de evaluación

Las componentes que reciben calificación son las siguientes:

- Ejercicios para resolver y entregar en cada clase. El profesor elegirá cinco de estos ejercicios para ser calificados. Cada ejercicio cuenta un 10 % de la nota definitiva.
- Asistencia a clase. La asistencia se verifica a partir de la entrega de los ejercicios hechos en clase. No entregar un ejercicio se asume como una falta a clase. Cada falta recibe una nota de 0.0 y cada asistencia recibe una nota de 5.0. El promedio de estas notas corresponde al 20 % de la nota definitiva. Esta contribución a la nota definitiva será de cero (0.0) si se dejaron de entregar **ocho** o más de estos ejercicios. Es necesario asistir a clase para que la entrega se tome como válida.
- Un examen final (con una componente escrita y otra de programación) con un valor del 30 % de la nota definitiva.

Al comienzo del semestre se hará un examen (5 % de bono sobre la nota definitiva) para diagnosticar el conocimiento general que ya tienen los estudiantes sobre los temas del curso.

VII Bibliografía

Bibliografía principal:

- (DABT) D.S. Sivia, J. Skilling, *Data Analysis. A Bayesian Tutorial*, Second Edition, 2012, Oxford.
- (ITILA) D. J. MacKay., *Information Theory, Inference and Learning Algorithms*, 2003, Cambridge.
<http://www.inference.phy.cam.ac.uk/mackay/itila/>.
- (ISLR) G. James, D. Witten, T. Hastie, R. Tibshirani., *An Introduction to Statistical Learning with Applications in R*, 2015, Springer.
<http://www-bcf.usc.edu/~gareth/ISL/>
- (ISTC) F. T. Wilmore, E. Jankowski, C. Colina, *Introduction to Scientific and Technical Computing*, 2017. CRC Press. (Biblioteca General - 502.85 I576)

Bibliografía complementaria:

- I. Goodfellow, Y. Bengio, A. Courville., *Deep Learning*, 2016, MIT.
<http://www.deeplearningbook.org/>
- A. Tveito, H.P. Langtangen, B.F. Nielsen., *Elements of Scientific Computing*, 2010. (Biblioteca General, Recurso Electrónico 510.)
- R. L. Burden, J. D. Faires. *Numerical analysis*, 2011. (Biblioteca General - 519.4 B862 2011)
- O. Maimon and L. Rokach, *The Data Mining and Knowledge Discovery Handbook*, 2010. (Biblioteca General, Recurso Electrónico 006.312)
- M. Snir, *MPI : the complete reference*, 1996. (Biblioteca General, 004.35 M637)
- J. Sanders, E. Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*, 2010. (Biblioteca General - 005.275 S152)
- D. Conway and J. M. White. *Machine learning for hackers*, 2012.
- Theano Development. *Deep Learning Tutorial* <http://deeplearning.net/tutorial/>
- J. VanderPlas., *Python Data Science Handbook*, 2016, O'Reilly.
<https://github.com/jakevdp/PythonDataScienceHandbook>