Análisis de datos y aprendizaje máquina

Carlos Malanche

26 de enero de 2018

Facultad de Ciencias, UNAM

Detalles principales del curso

Temario

Evaluación

Bibliografía

Detalles principales del curso

Meta del curso

Conseguir que el estudiante se familiarice con el uso de la computadora para manejar correctamente datos recabados de experimentos, realizar de manera correcta análisis estadístico e inferencias sobre la información disponible en grandes cantidades (*Big Data*). Aprender a hacer predicciones sobre los datos obtenidos de una manera básica al menos, con métodos del aprendizaje máquina.

Requerimientos

Se asumirá que el estudiante ya ha utilizado **al menos un lenguaje imperativo de programación** (`C`, `C++`, `python`, `java`, `R`, `Fortran` siendo los más adecuados) pues esto hará la curva de aprendizaje un tanto menos vertical. El lenguaje a utilizar será `Python` para el análisis de datos, posiblemente `C++`.

Para seguir el curso es preferible tener equipo propio (Windows, Linux o MacOS, los tres funcionarán).

El curso está abierto no sólo a estudiantes de física, pero puede ser cursado por estudiantes de matemáticas, ciencias de la computación y actuaría.

Requerimientos

Los requerimientos en términos de conocimiento son los siguientes:

- Álgebra lineal
- Cálculo diferencial en varias variables

No necesario, pero de gran ayuda:

- Estadística y Probabilidad
- Algoritmos (diseño y análisis de complejidad)
- Programación (a un nivel intermedio).

Temario

Cronología

El programa tendrá una duración de 16 semanas, 3 horas semanales divididas en una sesión teórica de 1 hora y 2 horas de laboratorio. El curso se va a repartir en cuatro bloques principales: Repaso de las herramientas computacionales, Análisis de datos, Métodos de regresión y clasificación, y Aprendizaje-máquina.

En porcentajes del curso, los bloques ocuparán 10 %, 30 %, 30 % y 30 % respectivamente.

Los temas en azul a lo mejor no se incluirán en el curso.

Bloque 1: Repaso de herramientas computacionales

- Repaso de bash (instrucciones *elementales*)
- Sistemas de control de versiones y comandos básicos de `git`.
- **Repaso** de C++ como (lenguaje de alto desempeño). Todas las instrucciones básicas (`for`,`while`,`if`, `switch`) más cosas *esenciales* de la programación orientada a objetos (variables y funciones miembro, modificadores de acceso, *herencia*).
- Análisis de algoritmos; complejidad algorítmica temporal.
- Cómputo de alto desempeño: Algoritmos de ordenamiento, estructuras básicas de datos (*stacks*, filas, árboles, listas).

Bloque 2: Análisis de datos

- **Repaso** de Python como lenguaje para *prototipos* de algoritmos de análisis de datos. Uso de la biblioteca `pandas`; estructuras de datos, lectura y escritura de archivos.
- Manejo de *Outliers* e información faltante.
- Interpretación estadística de datos.
- Representación visual de la información.
- El *p-value*.
- Ejemplos aplicados a sistemas físicos.

Bloque 3: Métodos de regresión y clasificación

- Fundamentos
  - Problemas inversos
  - Funciones de costo
  - Optimización
- Métodos de regresión y clasificación básica:
  - Regresión lineal
  - Regresión logística
  - Regresión de Tikhonov (en algunos lados *regresión de arista*)
  - k-Vecinos más cercanos (*k-NN*)
- Detalles de la regresión
  - Sobreajuste (*Overfitting*)
  - Validación cruzada (*Cross-validation*)

Bloque 4: Aprendizaje-máquina

- Análisis de *clusters* (Aprendizaje sin supervisión).
- Máquinas de vectores de soporte (SVMs).
- Redes neuronales.
- Regresión de procesos Gaussianos (*GPR*, *métodos de kernel*)
- `Scikit-learn` en `Python` para el aprendizaje-máquina.
- Ejemplos aplicados a sistemas físicos.

Evaluación

Evaluación

La primera componente de la calificación será un examen final, cuya calificación detonaré con E_f.

Durante el curso, habrá n tareas (a la mejor cada 2 semanas) que podrán aportar hasta 2 puntos sobre la calificación final bajo la condición de obtener **como mínimo** $E_f = 6$. Si la calificación de la tarea número i es denotada como t_i, entonces la calificación final será:

$$\text{calif}_{final} = \text{mín}\{10, \quad E_f + (\frac{1}{5n}\sum_{i=1}^{n} t_i)\mathbb{1}_{E_f \geq 6}\}$$

Habrá segunda vuelta de final, pero la nota será **la obtenida en la segunda vuelta únicamente**.

Evaluación

No se busca que el estudiante sepa todo el temario de memoria, sólo que cuente con las habilidades necesarias para hacer conclusiones *razonables* frente a un gran conjunto de datos, y eso es lo que el examen final evaluará.

Más detalles

Las tareas del primer bloque serán en `C++`, sugerido utilizar gcc 7.2.0 (disponible para Unix y para Windows bajo el nombre MinGW), Visual Studio Code como editor. Yo propocionaré ayuda para configurar computadoras personales si tienen problema con ello. Adicionalmente para Windows, se recomienda el uso de `Cmder` para emulación de bash en Windows (Windows 10 ya tiene bash, pero es un tanto complicado). Las tareas del tercer y cuarto bloque serán a hacer en iPython notebook, y usarán los paquetes `numpy`, `pandas` y `scikit-learn`. Si el tiempo lo permite, usaremos también OpenNN Se abrirá un *repo* de git especial para todas las tareas.

Bibliografía

Bibliografía

Lamentablemente la bilbiografía está toda en inglés.

- Christipher Bishop, *Pattern Recognition and Machine Learning*, Springer 2006.
- C. E. Rasmussen & C. K. I. Williams, *Gaussian Processes for Machine Learning*, the MIT Press, 2006.
- Charles M. Grinstead & J. Laurie Snell, *Introduction to Probability*, Dartmouth College.