

Redes: Programa

Leo Ferres, PhD, UDD/IDS, ISI, TEF, lferres@udd.cl Santiago de Chile, Dec 2019

[v.2019-12-06 16:39:00 -0300]

Resumen y objetivo general

Grafos como árboles y redes son estructuras de datos fundamentales que han dado origen a las Ciencias de la Complejidad, una de las disciplinas más originales y fecundas de la última década. Sus resultados subyacen muchos de los recientes avances de Ciencias de Datos y Ciencias de la Computación. Las aplicaciones de esta disciplina van desde la biología hasta la interacción humana, pasando por tecnologías como las redes sociales, la computación distribuida, el camino más corto entre dos puntos en un mapa, etc., donde todo se apoya en el uso apropiado de los conceptos de teoría de grafos y redes.

En este curso en particular, los estudiantes construirán, paso a paso, un conjunto de herramientas conceptuales (representación y algoritmos de redes) y prácticas (como bibliotecas de funciones y software), que se usan para implementar el estado del arte en algoritmos tales como PageRank de Google, bases de datos de grafos, y otras herramientas como web crawlers. Estas herramientas teóricas y prácticas les permitirán a los alumnos, a través de tareas y un proyecto final, a realizar sus propios análisis de datos de redes reales

Admin

- Profesor: Leo Ferres, PhD (lferres@udd.cl)
- TA: Pedro Avila (p.avila@udd.cl)
- Sala y horarios:

FECHA	HORA	ASIGNATURA	HRS. ACADÉMICAS	SUMA HRS. ACADÉMICAS
Sábado 07 Diciembre 2019	08.50 - 10.10	ELECTIVO III	2	2 de 20
	10.10 - 11.30	ELECTIVO III	2	4 de 20
	11.30 - 11.40	Coffee		
	11.40 - 13.00	ELECTIVO III	2	6 de 20
	13.10 - 14.30	ELECTIVO III	2	8 de 20
Sábado 21 Diciembre 2019	08.50 - 10.10	ELECTIVO III	2	10 de 20
	10.10 - 11.30	ELECTIVO III	2	12 de 20
	11.30 - 11.40	Coffee		
	11.40 - 13.00	ELECTIVO III	2	14 de 20
	13.10 - 14.30	ELECTIVO III	2	16 de 20
Sábado 11 Enero 2020	09.00 - 10.20	ELECTIVO III	2	18 de 20
	10.20 - 11.40	ELECTIVO III	2	20 de 20
	11.40 - 11.50	Coffee		

-
- Horas de consulta:

- Viernes de 5pm a 6pm (LF), o por cita vía email
- Comunicación:
 - **github**: https://github.com/leoferres/redes_msc_ids
- Material:
 - Apuntes del curso
 - Libros guía (para profundizar los temas)
 1. Network Science, Albert-László Barabási (<http://networksciencebook.com/>).
 2. Networks - an Introduction, Mark Newman, OUP
 3. Software: Networkx
 4. Datasets, los de **nx** entre otros, pej Barabási

Organización

- **Clases**: Las clases están divididas en *expositivas* (de “pizarra”) e introducen conceptos y teoría, *prácticas* (de “programación”), y *evaluativas* (presentaciones, tareas, ejercicios, etc.). Las clases son de asistencia obligatoria, en particular dado que perder una clase con los horarios de este curso implica perder 40% de los contenidos!
- **“Problem sets”**: Los “problem sets” son guías de ejercicios a ser resueltas en clase durante las sesiones prácticas o evaluativas (o a veces si no hay tiempo, completadas en la casa). Consisten en ejercicios matemáticos o de programación acorde a los temas tratados ese día. Al final del curso, se pedirán estas guías impresas (con nombre y apellido, como las tareas, ver más abajo).
- **Tareas**: Habrá 2 tareas (entre las clases 1 y 2, y 2 y 3) que se enfocarán en consolidar los conocimientos de la clase anterior a más profundidad basada en un trabajo más acorde al ideal de enseñanza-aprendizaje que el de la clase. Se espera que los alumnos entreguen un **pdf** de la tarea, y suban el notebook online, junto con los datasets, etc. para poder evaluar el código. Habrá más información de las tareas en un “handout” aparte.
- **Evaluación final**: Una evaluación integrativa de las clases 1 y 2, de 45 minutos, durante la clase 3 *en sala*, con papel y lapiz.
- **Proyecto**: El proyecto final es la parte más compleja de este curso y está dividida en 2 partes. La primera es un “pitch” de estrictamente tres minutos por alumno en el que presentan **una** slide en la clase 3 sobre la motivación, el dataset, la metodología, y los resultados esperados. Algunas precisiones:
 1. *Datasets*: nosotros podemos sugerir datasets, o ustedes pueden proponer otros, pero asegúrense de chequear con el staff del curso si los datasets se ajustan a las guías/expectativas que los profesores tienen del curso. Pregunten!

2. *Ayuda*: El proyecto es el componente más importante de la nota (30% + 10% del pitch). Tienen que empezar a pensar en él lo antes posible. El proyecto tiende a ejemplificar un proyecto real de investigación en data science (a escala, por supuesto), y por lo tanto toma tiempo y esfuerzo. Estas cosas nunca “funcionan a la primera”: bugs, análisis equivocados, narrativa incoherente, etc. A todos nos pasa lo mismo! Así que no esperen al último minuto para hacer el proyecto... se va a notar.

Más info en un “handout” aparte sobre el proyecto.

Las calificaciones finales del curso estarán basadas en los cinco componentes mencionados más arriba:

1. Tareas: 30% (15% cada una)
2. Presentación intermedia del proyecto: 10%
3. Proyecto: 30%
4. Problem sets: 10%
5. Evaluación final: 20%
6. Asistencia a clase: 5%

Sí, la suma es 105%. Ese 5% es importante para subir la nota del examen, por ejemplo, o de las tareas. Si un estudiante obtiene una calificación de 105%, quedará como 100% (un 7), usted ya es lo máximo que se puede ser en este curso.

Políticas del curso

[Inspirado en mis cursos, el “syllabus” de A-L Barabási, Erik Demain, y las Aha! Sessions de Don Knuth]

Ciencias de la computación y matemática

[Inspirado en el “syllabus” de A-L Barabási, pero con contenido esencialmente diferente]

“Network Science” (ciencia de redes, o sistemas complejos) es fundamentalmente una disciplina computacional, y al mismo tiempo, las matemáticas son inevitables para hacer justicia a la disciplina. Sin embargo, por las características de este curso (el tiempo, y la variedad de perfiles que usualmente existe) no podremos revisar *todos* los principios matemáticos fundamentales, ni técnicas de programación avanzadas. Este curso es, por necesidad, un híbrido entre ciencia, e ingeniería, pero ni lo uno, ni lo otro. Está pensado para alumnos que ya hayan cursado y sepan algo de Python (nuestros ejemplos estarán demostrados en código) y tengan un mínimo de conocimiento de matemática discreta y probabilidad. Pero **no se supone ningún conocimiento avanzado previo**. De

hecho, el curso está más dedicado a la interpretación intuitiva de las métricas y algoritmos de redes que a la enseñanza de “primeros principios”.

Colaboración

[Adaptadas de mi curso de programación de la UdeC en 2015, y a su vez this set of rules has been modified to fit this particular course from the original, created by E. Demain for his 6.046 course at MIT]

El objetivo de las tareas es que practiquen el material del curso. Entonces, se incentiva a que colaboren y hagan las tareas en grupo. Si trabajan en grupo, sin embargo, tienen el deber consigo mismo y con los otros miembros del grupo de estar preparados para la reunión del grupo. En principio, antes de dicha reunión, deberían pasar una hora tratando de resolver el problema solos. Si el grupo no puede resolver el problema, pueden hablar con otros grupos, o con los ayudantes, o postear dudas específicas online.

Sin embargo, **el documento de explicación del código deben escribirlo solos, sin ninguna clase de ayuda de parte de nadie más.** Aún si colaboraron con alguien más. Es importante que identifiquen con quién trabajaron en grupo, si lo hizo sólo, solamente coloque “colaboradores: ninguno”. Si tomaron algo de la web, por favor también den crédito a la fuente, en la mejor tradición escolástica. Así y todo, tienen que escribir la solución de la tarea en sus propias palabras. **Ustedes deben ser capaces de explicar el código con sus propias palabras al staff del curso.** El plagio y otra conducta deshonestas no será tolerada en la universidad. Si tienen preguntas sobre la política de colaboración del curso, por favor pregunten por Piazza. Si tienen dudas de haber violado la política de colaboración, por favor acérquense a hablar con el staff. Si bien los casos de plagio y copia deben ser manejados severamente, usualmente somos más comprensivos si ustedes nos cuentan que si nos enteramos por terceros.

El examen final es individual. No se permite la colaboración.

Cambios de contenido y responsabilidad de los estudiantes

El profesor se reserva el derecho de cambiar partes sustanciales de este syllabus. Los cambios se discutirán con los alumnos durante la clase o por email. Los estudiantes son los responsables de la información dada durante las clases. No asuman algo simplemente porque no está especificado en este handout. Si tienen preguntas, por favor consulten con el staff.

Contenidos generales

Los contenidos de este curso están basados fuertemente en los libros mencionados mas arriba y el curso de redes de A-L Barabási, el excelente tutorial de Bruno Goncalves, una serie de blog posts de Maël Fabien, que considero del nivel correcto para este magister (una mezcla de teoría y práctica), el curso de Chato Castillo en la UPF y en las notas y clases de mi predecesor en este curso, Cristian Candia (basadas también en el curso de Barabási).

1. Clase 1 - Intro y formación de redes I
 1. Admin (este documento)
 2. Motivación
 3. Fundamentos matemáticos de teoría de grafos (redes)
 4. Fundamentos computacionales (representación)
 5. Paths
 6. ...
 7. Código
2. Clase 2 - Formación de redes II
 1. Modularidad
 2. Comunidades
 3. ...
 4. Código
3. Clase 3 - Redes dinámicas: Modelos
 1. Strogatz
 2. Barabási-Albert
 3. ...
 4. Código

Metodología pedagógica

El uso intensivo del pizarrón es una decisión consciente. Les permite a los estudiantes:

- pensar preguntas mientras se escribe
- tener tiempo para tomar notas
- saber qué es crítico de lo que se está diciendo en clase
- saber que el instructor también tiene que “trabajar” en escribir
- procesar lo que se dice, en mi opinión, mucho más que con Powerpoint.
- “seguir” al profesor sin tener que mirar las slides/diapos

Además, trabajar con pizarra tiene un tinte “old school” que entretiene, en nuestra opinión, y que hace más fácil luego seguir los videos por la web.

Toda la práctica se dará en clases, y los códigos demostrados en los labs y en clases estarán disponibles en github. Se espera que el estudiante transcriba, descargue, analice y ejecute estos códigos de ejemplo porque muchos contienen varias lecciones que vale la pena aprender.

Filmación del curso

En el 2019, el staff tiene como objetivo filmar el curso.

La cámara está usualmente apuntando al instructor del curso, y los alumnos no debieran aparecer en la filmación. Sin embargo, si alguno tiene problemas en aparecer en la filmación, por favor sentarse atrás en la sala.

Los videos deberían aparecer durante la misma semana de la filmación en un link a ser difundido por Piazza y el mail del grupo.

```
pandoc -o programa.pdf programa.md
```