

### Formulario de aprobación de curso de posgrado/educación permanente

**Asignatura: Aprendizaje Automático para Datos en Grafos**

(Si el nombre contiene siglas deberán ser aclaradas)

<b>Modalidad:</b> (posgrado, educación permanente o ambas)	<b>Posgrado</b>	<input checked="" type="checkbox"/>
	<b>Educación permanente</b>	<input checked="" type="checkbox"/>

---

**Profesor de la asignatura <sup>1</sup>:** Paola Bermolen (gr4, IMERL)

(título, nombre, grado o cargo, instituto o institución)

**Profesor Responsable Local <sup>1</sup>:**

(título, nombre, grado, instituto)

**Otros docentes de la Facultad:** Marcelo Fiori (gr3, IMERL), Federico La Rocca (gr4, IIE), Bernardo Marengo (gr2, IMERL)

(título, nombre, grado, instituto)

**Docentes fuera de Facultad:** Gonzalo Mateos (Associate Professor, University of Rochester, NY, USA)

(título, nombre, cargo, institución, país)

<sup>1</sup> Agregar CV si el curso se dicta por primera vez.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

**Programa(s) de posgrado:** Maestría en Ingeniería Matemática (IngeMat), Maestría en Ciencia de Datos y Aprendizaje Automático (MCDAA), Maestría y Doctorado en Ingeniería Eléctrica

**Instituto o unidad:** IMERL/IIE

**Departamento o área:**

---

**Horas Presenciales: 60**

(se deberán discriminar las horas en el ítem Metodología de enseñanza)

**Nº de Créditos: 10**

[Exclusivamente para curso de posgrado]

---

(de acuerdo a la definición de la UdelaR, un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem Metodología de enseñanza)

**Público objetivo:** Estudiantes avanzados de grado o de posgrado con formación básica en álgebra lineal y probabilidad y cuyos intereses estén relacionados con el modelado, análisis estadístico y/o o visualización de datos en grafos. El curso espera llegar a estudiantes de programas de la Facultad de Ingeniería (IngeMat, MCDAA, etc) pero también a estudiantes de posgrado por fuera de la facultad como la maestría en Bioinformática, maestría en Ciencias Cognitivas, maestría en Ecología y Evolución o maestría en Ciencias Ambientales. Cada estudiante podrá profundizar en alguno de los aspectos de su interés: modelado matemático, análisis descriptivo, visualización, análisis estadístico etc.

**Cupos:** No tiene cupos

---

### Objetivos:

Los grafos (o redes) son una estructura de datos presente en muchísimas áreas de conocimiento: redes de telecomunicaciones, sistemas de recomendación, redes de regulación genética, estructuras de proteínas o movilidad urbana son solo algunos ejemplos. Básicamente, son entidades (nodos) que interactúan entre sí (aristas).

Sobre estos datos existen muchos problemas interesantes de aprendizaje automático, donde básicamente se busca realizar predicciones o descubrir cierta estructura en los datos: detección de anomalías en una red inalámbrica, recomendación de libros a partir de clasificaciones anteriores y de otros usuarios, o clasificación del rol de ciertas proteínas en redes de interacción biológica. Sin embargo, está claro que técnicas de aprendizaje “tradicionales” (donde los datos son básicamente un vector o una matriz) que no toman en cuenta las relaciones entre las distintas entidades tendrán menor poder de generalización (y por lo tanto mucho peor desempeño) que aquellas que sí lo tomen en cuenta.

El objetivo general del curso es que los estudiantes puedan afrontar un problema de aprendizaje automático donde los datos se encuentran en forma de grafos. Se brindarán los conceptos teóricos fundamentales y las herramientas prácticas necesarias para ello. Al finalizar el curso los estudiantes serán capaces de implementar y entender distintas técnicas del estado del arte en inferencia y predicción en grafos.

---

**Conocimientos previos exigidos:** Álgebra de matrices. Fundamentos de probabilidad y estadística Programación a nivel básico.

**Conocimientos previos recomendados:** Conocimientos básicos en fundamentos de aprendizaje automático. Programación en python científico.

---

**Metodología de enseñanza:**

(comprende una descripción de la metodología de enseñanza y de las horas dedicadas por el estudiante a la asignatura, distribuidas en horas presenciales -de clase práctica, teórico, laboratorio, consulta, etc.- y no presenciales de trabajo personal del estudiante)

Descripción de la metodología:

El curso consta de dos clases teóricas presenciales durante 14 semanas además de seis clases prácticas asociadas a cada uno de los temas del curso.

Las clases de práctico incluyen un taller práctico en python que busca ilustrar algunos de los conceptos vistos en clase. Éstos serán realizados por los estudiantes y servirán sobre todo para auto-estudio y preparación del trabajo final. El objetivo es que luego de realizar el taller el estudiante sea capaz de levantar un conjunto de datos y aplicarle algunas de las técnicas que se verán en el curso, independientemente del trabajo final que elija.

Detalle de horas:

- Horas de clase (teórico): 48
- Horas de clase (práctico): 12
- Horas de clase (laboratorio):
- Horas de consulta:
- Horas de evaluación:
  - o Subtotal de horas presenciales: 60
- Horas de estudio: 20
- Horas de resolución de ejercicios/prácticos: 20
- Horas proyecto final/monografía: 50
  - o Total de horas de dedicación del estudiante: 150

---

**Forma de evaluación:** Entrega de ejercicios prácticos y/o en el lenguaje de programación *python* (una entrega por cada clase de práctico) junto con un proyecto final.

En el proyecto final, los estudiantes podrán investigar y aplicar algoritmos de aprendizaje automático en grafos del estado del arte, a una aplicación de su interés.

Una vez concluido el componente instructivo del curso y realizada la entrega de ejercicios, los estudiantes deberán seleccionar un tema específico, relacionado con el material tratado en clase, y realizar un estudio relativamente profundo del tema. Esto implica por ejemplo buscar fuentes de literatura (trabajos de investigación, libros de texto, etc.), y eventualmente realizar algunos análisis y/o simulaciones numéricas para experimentar con conjuntos de datos sobre grafos, y proporcionar un resumen detallado de las ideas principales en formato paper.

---

### Temario:

1. Introducción, motivación y desafíos del área.
2. Teoría básica de grafos
  1. Preliminares: representación, grafos dirigidos y con pesos, árboles, grados, matriz de adjacencia y matriz laplaciana
  2. Métricas y medidas: centralidad, transitividad y coeficiente de clustering, similaridad.
3. La estructura de las grandes redes de datos
  1. Componentes, caminos más cortos (small world effect), redes con distribución power-laws y scale-free, assortative mixing
  2. Partición del grafo: partición espectral, modularidad, otros métodos.
4. Modelos de grafos: random graphs (Erdos-Renyi), configuration model, preferential attachment, random dot product graphs (RDPG).
5. Inferencia en grafos: El problema de inferencia de topología.
  1. Preliminares de AA (inferencia, regresión lineal)
  2. Métodos estadísticos (grafos de correlación, selección de covarianza con graphical lasso, y regresión basada en vecinos). Aprendizaje de grafos a partir de observaciones de señales: análisis basados en Laplaciano, suavidad y esparsidad. Identificación de estructura en procesos de difusión. Aplicación a redes de regulación genética, estructura de proteínas, y patrones de movilidad urbana.
6. Graph Neuronal Networks (GNN):
  1. Conceptos básicos de GSP (graph signal processing): graph-shift operator; graph Fourier transform (GFT); y filtros/convoluciones en grafos.
  2. Arquitecturas de GNN como composición de filtros en grafos con no-linealidades. Grafos en el dominio del espectro; ChebNet; CayleyNet; redes convolucionales en grafos.

Propiedades fundamentales de las GNN: invariancia a permutaciones, estabilidad a deformaciones de los grafos, expresividad y poder de discriminación. Variantes: graph attention networks; recurrent GNNs; pooling. Aplicaciones a predicción de aristas, aprendizaje semi-supervisado y sistemas de recomendación.

7. Estimación y Clasificación para datos en grafos: Detección de comunidades, Laplacian embeddings y spectral clustering. Node y (sub)graph representation learning: métodos basados en factorización de matrices, paseos aleatorios, y redes convolucionales en grafos. Aplicaciones a clasificación de nodos y de grafos, y modelado de la relación entre conectividad estructural y funcional en el cerebro.

---

### Bibliografía:

- Statistical Analysis of Network Data Methods and Models, Eric Kolaczyk, Springer, 2009.
- Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016. Disponible en línea: <https://www.deeplearningbook.org/>.
- Ortega, P. Frossard, J. Kovačević, J. M. F. Moura and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," in Proceedings of the IEEE, vol. 106, no. 5, pp. 808-828, May 2018.
- G. Mateos, S. Segarra, A. G. Marques and A. Ribeiro, "Connecting the Dots: Identifying Network Structure via Graph Signal Processing," in IEEE Signal Processing Magazine, vol. 36, no. 3, pp. 16-43, May 2019.
- X. Dong, D. Thanou, M. Rabbat and P. Frossard, "Learning Graphs From Data: A Signal Representation Perspective," in IEEE Signal Processing Magazine, vol. 36, no. 3, pp. 44-63, May 2019.
- M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean data," in IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 18-42, July 2017.
- J. Zhou et al, "Graph Neural Networks: A Review of Methods and Applications", arXiv:1812.08434v4 [cs.LG].
- F. Gama, E. Isufi, G. Leus, A. Ribeiro, "Graphs, Convolutions, and Neural Networks," arXiv:2003.03777v1 [cs.LG].
- T. N. Kipf, M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," arXiv:1609.02907v4 [cs.LG].
- W. L. Hamilton, R. Ying, J. Leskovec, "Representation Learning on Graphs: Methods and Applications," arXiv:1709.05584v3 [cs.SI].



- T. N. Kipf, M. Welling, Variational Graph Auto-Encoders, NIPS Workshop on Bayesian Deep Learning (2016).
  - J. You, R. Ying, X. Ren, W. L. Hamilton, J. Leskovec, “GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Model”, ICML 2018.
  - William L. Hamilton, “Graph Representation Learning”, Morgan & Claypool, 2020. Borrador disponible en línea: [https://www.cs.mcgill.ca/~wlh/grl\\_book/](https://www.cs.mcgill.ca/~wlh/grl_book/).
  - A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, and Y. Qin, “Statistical inference on random dot product graphs: A survey,” J. Mach. Learn. Res., vol. 18, no. 1, p. 8393–8484, January 2017.
  - B. Marenco, P. Bermolen, M. Fiori, F. Larroca, G. Mateos, “On-line change point detection for weighted and directed random dot product graphs”, Transactions on Signal and Information Processing over Networks (TSIPN), DOI: 10.1109/TSIPN.2022.3149098.
-

**Datos del curso**

---

**Fecha de inicio y finalización:** Agosto a Diciembre de 2023

**Horario y Salón:**

**Arancel:**

[Si la modalidad no corresponde indique "no corresponde". Si el curso contempla otorgar becas, indíquelo]

**Arancel para estudiantes inscriptos en la modalidad posgrado:**

**Arancel para estudiantes inscriptos en la modalidad educación permanente:** 5000UI

**Actualizado por expediente n.º:** 060140-000014-23

---