Node label and Link prediction in complex networks

Miguel Romero

October 11, 2019

Pontificia Universidad Javeriana de Cali
Seminario Permanente de la Facultad de Ingeniería y Ciencias

Overview

1. Node label prediction:

In-Silico Gene Annotation Prediction using the Co-expression Network Structure

2. Link prediction:

Spectral Evolution of Twitter Mention Networks

Node label prediction: In-Silico Gene Annotation Prediction using the Co-expression Network Structure

Objective

In-silico prediction of functional gene annotations.

How?

- gene co-expression network
- existing knowledge body of gene annotations of a given genome
- supervised machine learning model

Co-expression networks

Co-expression networks are generally, represented as undirected, weighted graphs built from empirical data (expression profiles). Vertices denote genes and edges indicate a weighted relationship about their co-expression.

Definition 1

Let V a set of genes, E a set of edges that connect pairs of genes and w a weight function. A *(weighted) gene co-expression network* is a weighted graph $G = (V, E, w : E \to \mathbb{R}_{\geq 0})$.

Gene annotation

The goal of **gene annotation** is to determine the structural organization of a genome and discover sets of gene functions, i.e., the locations of genes and coding regions in a genome that determine what genes do.

Gene annotation

Gene annotations are classified in

- molecular function: molecular activities of individual gene products,
- cellular components: location of the active gene products,
- **biological processes**: pathways to which a gene contributes.

Gene annotation

Definition 2
Let A be a set of biological functions. An *annotated gene co-expression network* is a gene co-expression network $G = (V, E, w)$ complemented with an annotation function $\phi : V \mapsto 2^A$.

Topological Properties

Given $G = (V, E, w)$, properties of its network structure are computed for each gene $u \in V$:

- degree,
- eccentricity,
- clustering coefficient,
- closeness centrality,
- betweenness centrality,
- neighborhood connectivity,
- topological coefficient.

Network-based approach

Given an annotated co-expression network $G = (V, E, w)$ with annotation function ϕ, the **goal** is to use the information represented by ϕ together with topological properties of G to obtain a function $\psi : S \mapsto 2^A$.

Function ψ predicts associations between annotations and genes based on a supervised machine learning technique.

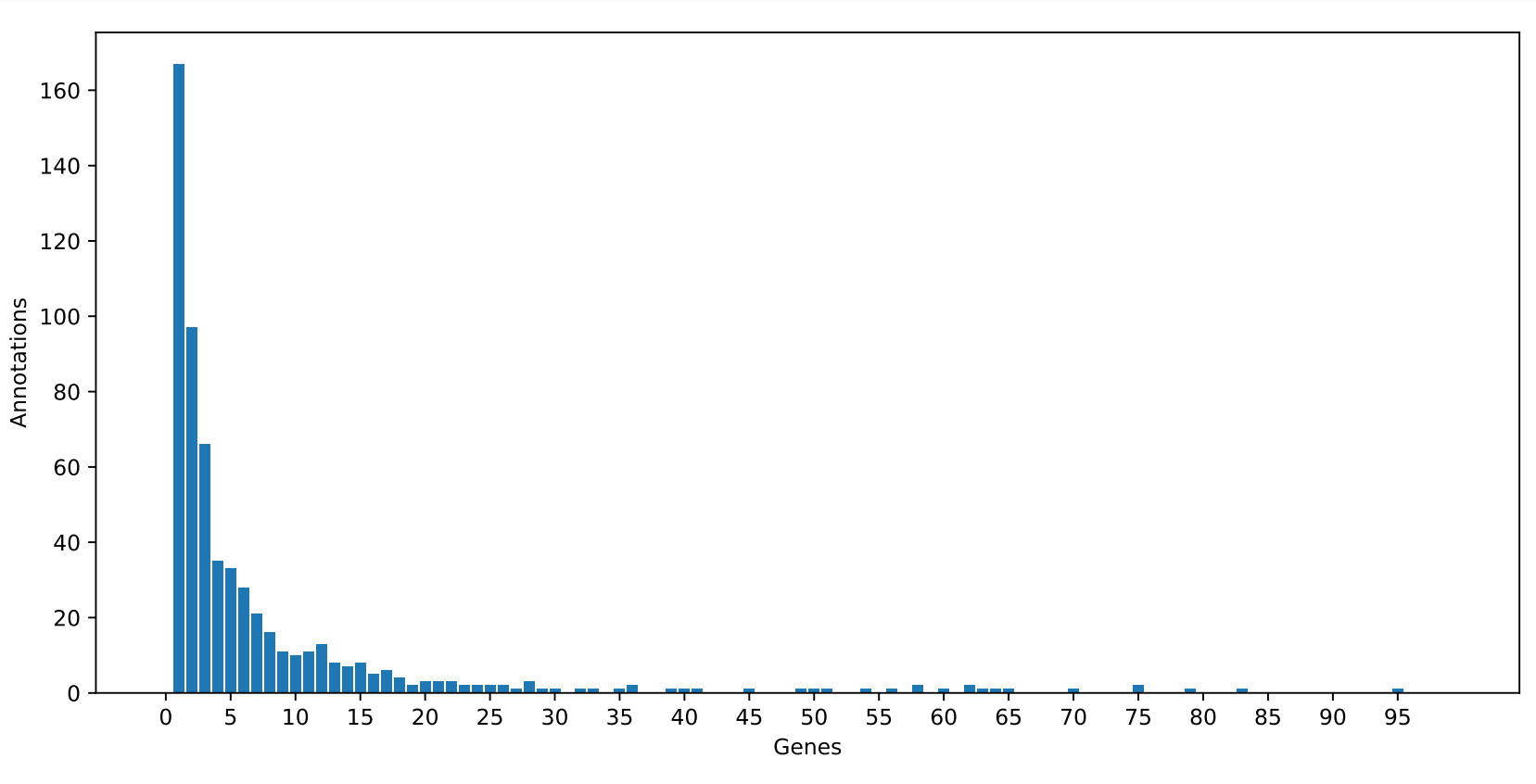
Oryza Sativa

The gene co-expression network $G = (V, E, w)$ comprises $19\,665$ vertices (genes) and $553\,125$ edges.

The dataset summarizes data for the $19\,665$ genes, 615 annotations, and 7 topological measures. It comprises $19\,665$ rows and 222 columns.

Supervised training

The dataset is heavily imbalanced!



Supervised training

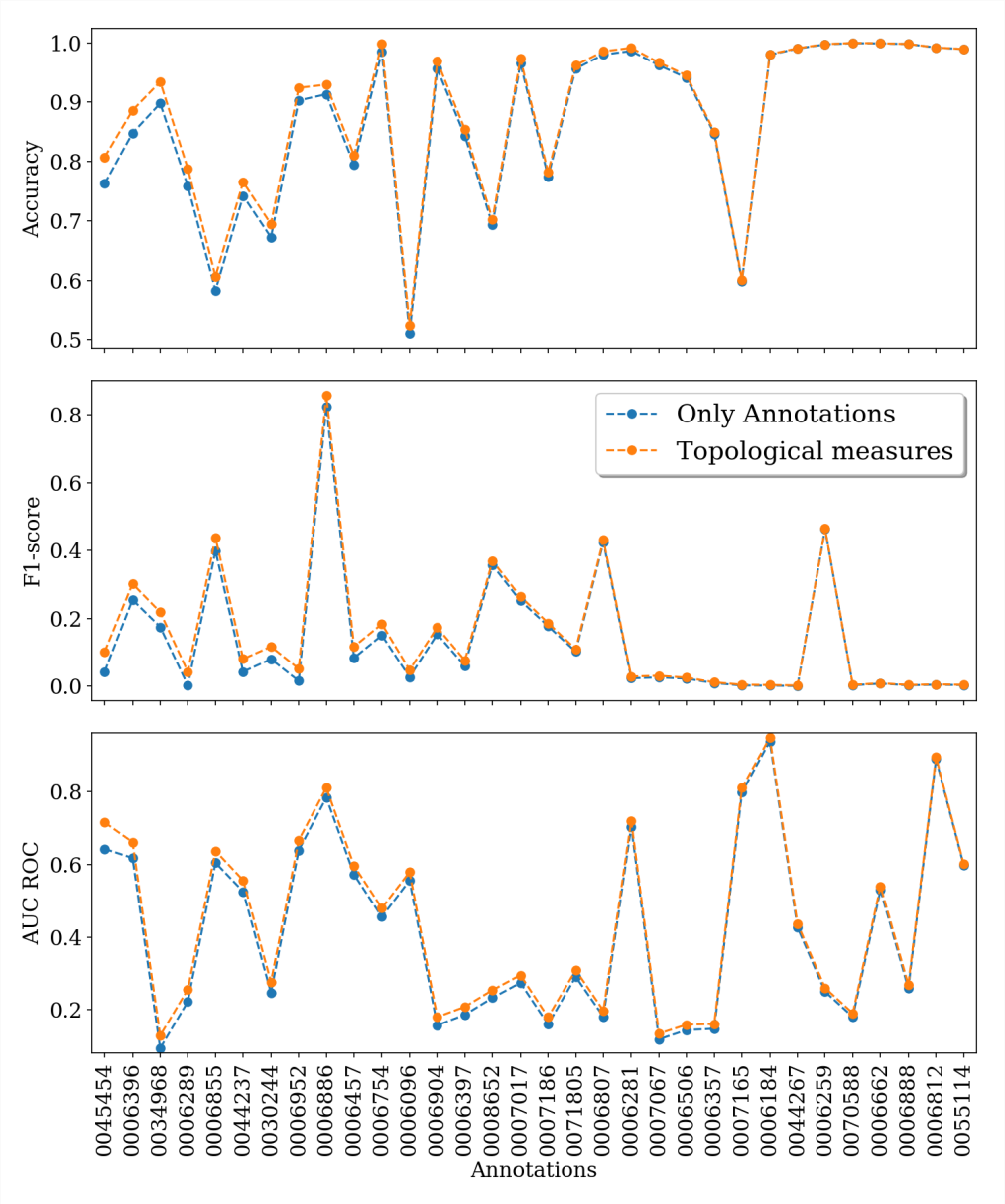
Synthetic Minority Over-sampling TEchnique (SMOTE) is used to over-sample the minority class to potentially improve the performance of a classifier without loss of data.

A supervised machine learning technique for the annotation prediction is used. In particular, the **XGBoost** implementation of gradient boosted decision trees is used.

Annotation prediction

Two models are trained for predicting gene annotations, one per biological function (141 annotations). Namely, one in which the topological measures of G are used and another one in which they are not.

Annotation prediction



Annotation prediction

ID	Biological process	# Genes	Max FP	# FP
0006807	nitrogen compound metabolic process	15	41	1
0006289	nucleotide-excision repair	20	46	1
0006397	mRNA processing	17	48	1
0007017	microtubule-based process	18	49	1
0070588	calcium ion transmembrane transport	10	36	1
0006184	GTP catabolic process	49	47	1
0044267	cellular protein metabolic process	25	49	1
0007186	G-protein coupled receptor protein signaling ...	11	50	1
0006281	DNA repair	62	50	2
0006754	ATP biosynthetic process	24	49	3
0006904	vesicle docking involved in exocytosis	11	50	4
0055114	oxidation-reduction process	870	47	5

Link prediction: Spectral Evolution of Twitter Mention Networks

Objective

Evaluate various link prediction methods that underlie the spectral evolution model.

Applies the spectral evolution model to networks of mentions between individuals who used trending political hashtags in Twitter between August 2017 and August 2018.

Mention networks

The dataset consists of 31 mention networks between Twitter users who defined their profile location as Colombia. These networks capture conversations around a set of hashtags related to popular political topics between August 2017 and August 2018.

Mention networks

Users are represented by the set of vertices V and the set of edges is denoted by E. There exists an edge $\{i, j\} \in V \times V$ between users i and j, if user i uses a political hashtag (e.g., #eleccionesseguras) and mentions user j (via @username).

A **mention network** $G = (V, E)$ is represented as a weighted multi-graph without self-loops. Our analysis is based on the largest connected component of the multi-graph, denoted by $G_c = (V_c, E_c)$.

Mention networks

	Hashtag	$\lvert V_c \rvert$	$\lvert E_c \rvert$
0	abortolegalya	1282	1538
1	alianzasporlaseguridad	150	351
2	asiconstruimospaz	2405	6950
3	colombialibredefracking	1476	3127
4	colombialibredeminas	655	1421
5	dialogosmetropolitanos	932	4134
6	edutransforma	161	404
7	eleccionesseguras	2634	7969
8	elquedigauribe	2052	5272
9	frutosdelapaz	1479	3468

Spectral evolution model

Let $\mathbf{A}$ denote the adjacency matrix of G_c. Furthermore, let $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ denote the eigen decomposition of $\mathbf{A}$, where $\mathbf{\Lambda}$ represents the spectrum of G_c.

The **spectral evolution model** characterizes the dynamics of G_c (i.e., how new edges are created over time) in terms of the evolution of the spectrum of the network, assuming that its eigenvectors in $\mathbf{U}$ remain unchanged.

Spectral evolution model

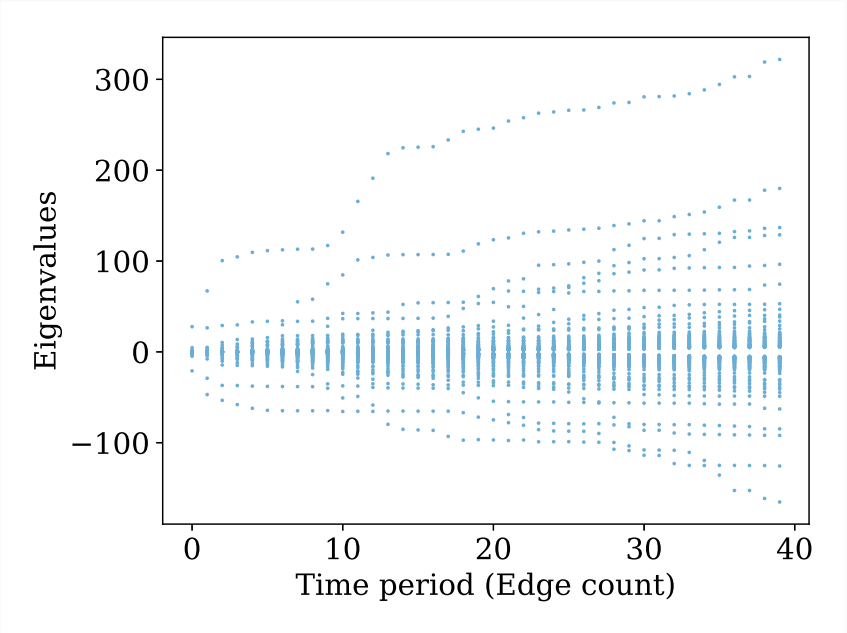
If this condition is satisfied, estimating the **formation of new edges** can be expressed as **transformations of the spectrum** through the application of real functions (using graph kernels) or extrapolation methods (using learning algorithms that estimate the spectrum trajectories).

Spectral evolution model verification

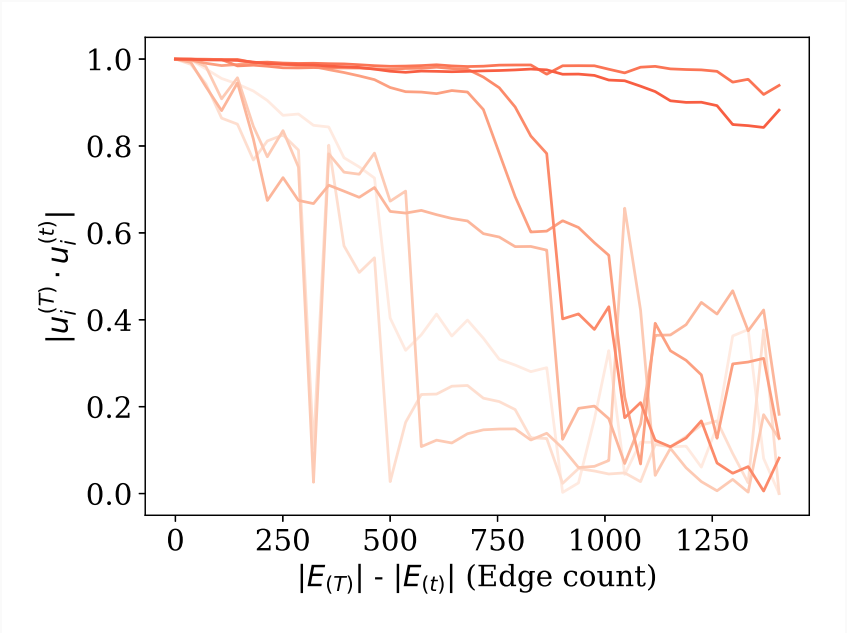
To apply the spectral evolution model, we need to verify the assumption on the evolution of the spectrum and eigenvectors. Every network G_c has a timestamp associated to each edge, representing the time at which the edge is created.

- spectral evolution (eigenvalues),
- eigenvector evolution,
- eigenvector stability, and
- spectral diagonality test

Spectral evolution model verification

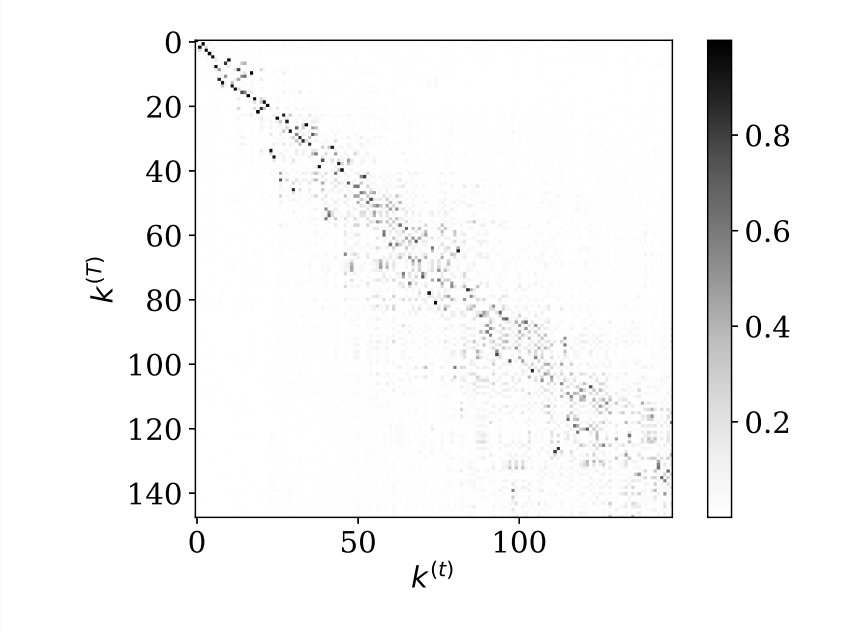


(a) Spectral evolution (eigenvalues)

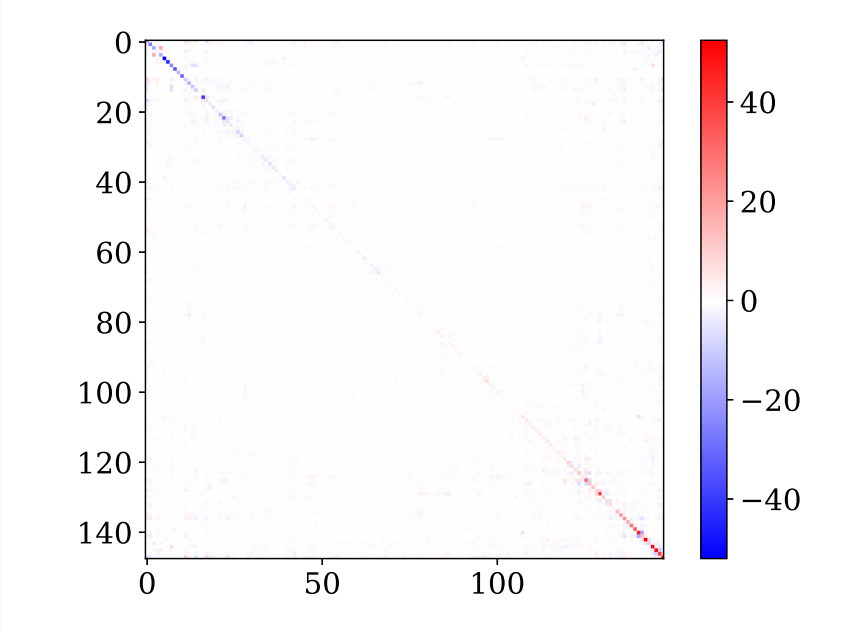


(b) Eigenvector evolution

Spectral evolution model verification



(c) Eigenvector stability (eigenvalues)



(d) Spectral diagonality test

Growth models

Let $K(\mathbf{A})$ be a kernel of an adjacency matrix $\mathbf{A}$, whose eigen decomposition is $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. **Graph kernels** assume that there exists a real function $f(\lambda)$ that describes the growth of the spectrum.

In particular, $K(\mathbf{A})$ can be written as $K(\mathbf{A}) = \mathbf{U}F(\mathbf{\Lambda})\mathbf{U}^T$, for some functions $F(\mathbf{\Lambda})$ that applies a real function $f(\lambda)$ to the eigenvalues of $\mathbf{A}$.

Graph kernels

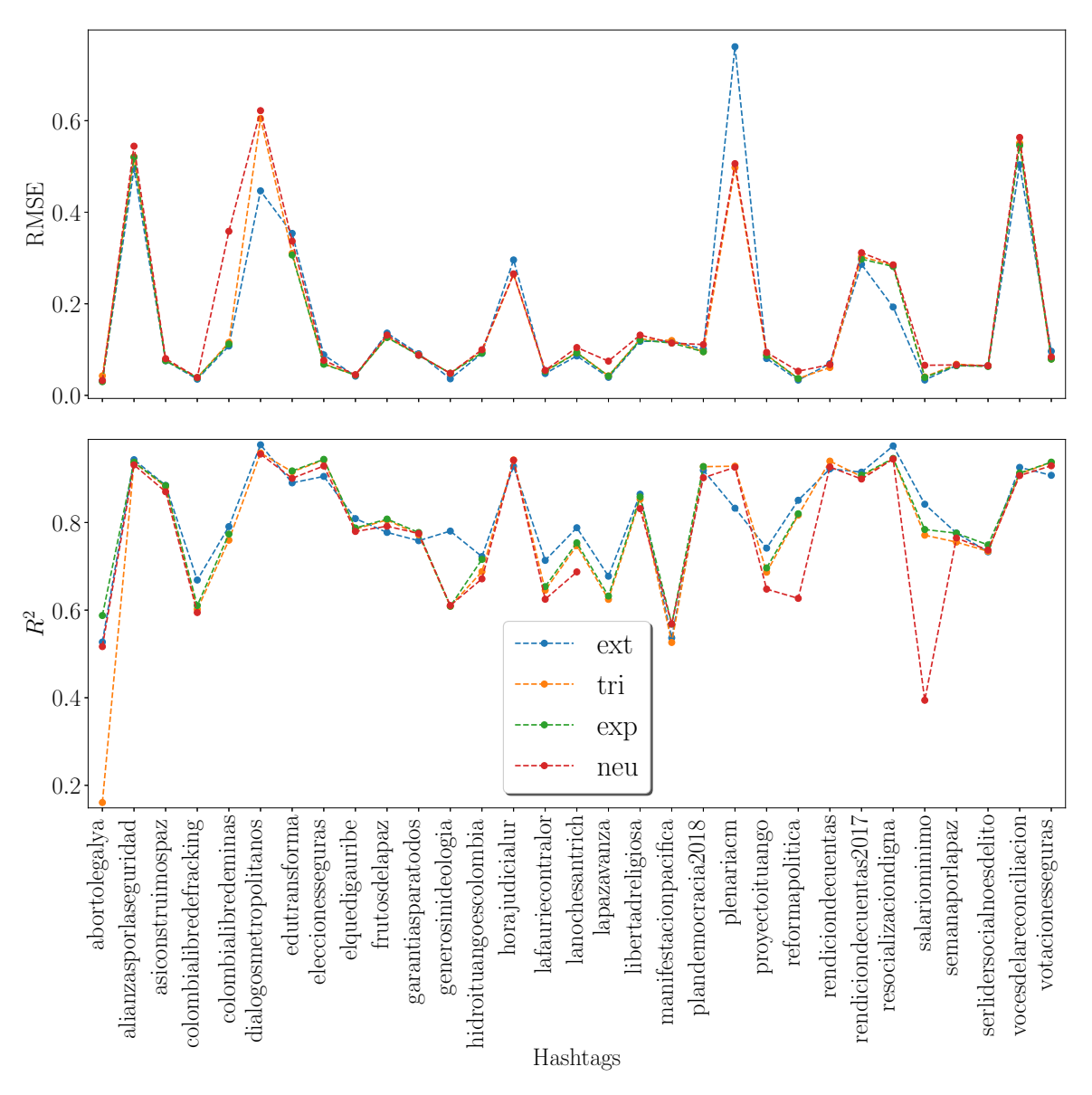
In particular, we use graph kernels of the triangle closing, exponential and Neumann growth.

Kernel	$K(\mathbf{A})$	$f(\lambda)$
Triangle closing	$\mathbf{A}^2$	$f(\lambda) = \lambda^2$
Exponential	$exp(\alpha\mathbf{A})$	$f(\lambda) = e^{\alpha\lambda}$
Neumann	$(\mathbf{I} - \alpha\mathbf{A})^{-1}$	$f(\lambda) = \frac{1}{1-\alpha\lambda}$

Spectral Extrapolation

When the evolution of the spectrum is irregular it is not possible to find a simple function that describe network growth. This model **extrapolates** each eigenvalue of the network, assuming that the network to be analyzed follows the spectral evolution model

Performance

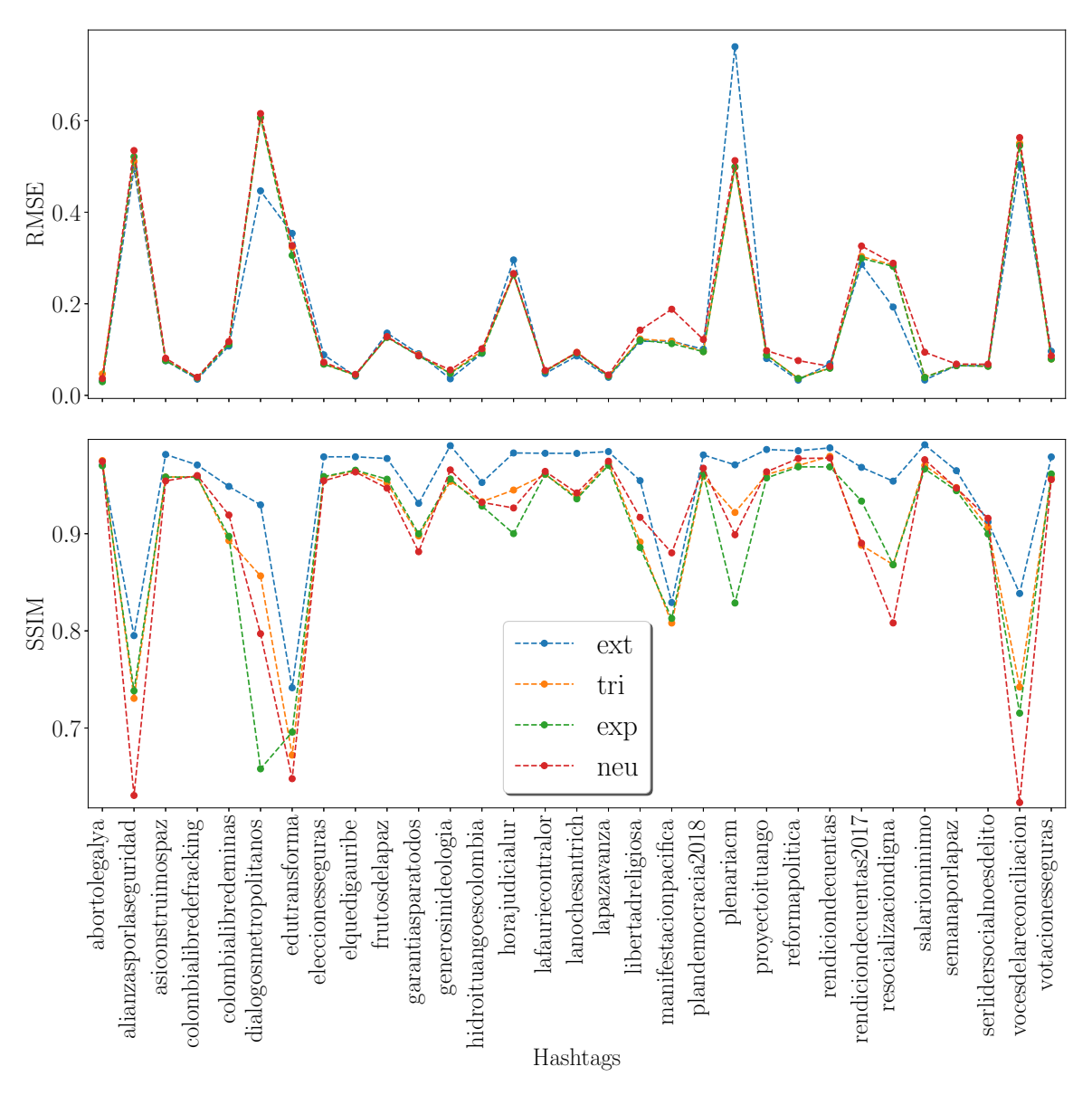


Structural similarity

Structural similarity (SSIM) is a method for measuring similarity between two images based on the idea that the pixels have strong inter-dependencies especially when they are spatially close.

Adjacency matrices $\mathbf{A}$ and $\hat{\mathbf{A}}_c$ are assumed to be images, and the edges $\mathbf{A}_{ij}$ and $\hat{\mathbf{A}}_{c,ij}$ represent pixels.

Performance



Conclusions

The extrapolation method tends to outperform the other methods based on the performance metrics. Specifically, for 28 out of 31 networks (91% of the total), the extrapolation method provides distinct, if slight, improvement.

The outperformance of the spectral extrapolation method seems to be explained by the method being able to consider the irregular evolution of the eigenvalues.

Questions?

Thanks!