Proyecto Capstone

“Pronóstico Híbrido Espacio-Temporal de Precipitaciones en Chile

Introducción

Los investigadores en climatología, especialmente aquellos dedicados a la predicción numérica del clima, la física atmosférica, eventos extremos y el cambio climático, están recurriendo cada vez más al aprendizaje automático (ML) para mejorar la modelación y predicción [4,5]. Sin embargo, las complejas correlaciones espaciales, temporales y espacio-temporales en los datos climáticos representan un desafío significativo para capturar y modelar patrones ambientales [6], en particular en el pronóstico de las precipitaciones.
El objetivo principal de este proyecto es desarrollar y perfeccionar técnicas de pronóstico para abordar la complejidad de los datos espaciotemporales utilizando redes neuronales en combinación con herramientas estadísticas, para crear métodos híbridos pragmáticos y parsimoniosos que sean eficientes en el pronóstico de variables climáticas, en particular – las precipitaciones. Estos métodos tendrían aplicaciones potenciales en diversos campos, incluyendo el clima, la hidrología, y la respuesta al cambio climático. Al mismo tiempo que se desarrollan las herramientas, estas son aplicadas al territorio nacional como un caso de estudio significativo, dado la gran variabilidad climática y la heterogeneidad geográfica de Chile.

Datos y Estrategias

Los datos que se utilizan provienen de ERA5 [7], que es parte del Servicio de Cambio Climático de Copernicus (C3S) proporcionado por la Unión Europea y producido por el Centro Europeo de Pronósticos Meteorológicos a Medio Plazo (ECMWF). ERA5 ofrece datos climáticos y meteorológicos reanalizados, cubriendo el período desde 1950 hasta el presente, proporcionando información detallada sobre una amplia gama de variables atmosféricas, terrestres y oceánicas. Su alta resolución espacial y temporal lo convierte en una herramienta ampliamente utilizada en investigaciones climáticas, estudios ambientales y aplicaciones de modelado meteorológico.
Los datos base [18][1] que serán utilizados en este proyecto forman una malla[2] de puntos ubicados entre las coordenadas geográficas de latitud -17.5° a -56.0° y longitud -76.0° a -66.0° con una resolución de 0.25×0.25 grados. Es decir, tenemos 41 valores de longitud y 155 valores de latitud, creando una malla de $n = 6355$ puntos espaciales. Cada punto está asociado con datos históricos que contienen series temporales de

[1] Los datos base llegan hasta el año 2022, para datos más actualizados hay que ir directamente al sitio de ERA5 y descargarlos.
[2] La malla para la Temperatura media superficial del mar es más grande, para incluir el territorio de Chile y el mar frente a su costa.

variables climáticas como temperatura máxima, media y mínima, precipitación, evapotranspiración, entre otras. Además, cada punto se caracteriza por sus coordenadas geográficas (longitud y latitud). Se tiene así en cada punto q = 26665 valores de tiempo, correspondientes a registros diarios entre 1980 y 2022.
Los datos descargados desde ERA5 tienen el formato **NetCDF**[3] (Network Common Data Form) con extensión "**.nc**" (Por ejemplo, el archivo de precipitaciones diarias - "ERA5_daily_precipitation_1950_2022.nc").
Para el análisis se sugiere estructurar los datos en la forma dada en la Tabla 1. Aquí $R_{t_i}^{r_j}$ denota el valor (mm) de la precipitación en la localización r_j, definida por r_j = $(longitud_j, latitud_j)$, en el tiempo t_i, con i =1 ...n, j=1 ...q.

	r_1	r_2	$\cdots$	r_q
t_1	$R_{t_1}^{r_1}$	$R_{t_1}^{r_2}$	$\cdots$	$R_{t_1}^{r_q}$
t_2	$R_{t_2}^{r_1}$	$R_{t_2}^{r_2}$	$\cdots$	$R_{t_2}^{r_q}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_n	$R_{t_n}^{r_1}$	$R_{t_n}^{r_2}$	$\cdots$	$R_{t_n}^{r_q}$

Tabla 1: Estructura sugerida para los datos.

En la predicción de las series de tiempo de precipitaciones se pueden aplicar directamente métodos Deep Learning (DL), que por lo general tienen mejor desempeño que los clasificadores superficiales (por ejemplo: ARIMA, PROPHET, GLMNET, SVM–RBF, BOOST–TREE (H2O), PROPHET–XGBOOST, RANDOM–FOREST, entre otros). Para comparar la estrategia fundamental del proyecto, que será propuesta más adelante, con la aplicación directa de métodos DL como base comparativa (baseline) se recomienda usar, con los datos dados, los recursos incluidos en la librería de Python **DeepAR** [9], que contiene numerosos métodos de vanguardia para el pronóstico probabilístico de series de tiempo. En particular, se sugiere usar[4]: DEEPAR, DEEP STATE, NBEATS, GAUSSIAN PROCESS FORECAST, etc. Para comparar el desempeño utilizar las métricas clásicas: MAE, MAPE, MASE, SMAPE, y RMSE.
Aplicar DL directamente sobre las series de precipitación tiene, sin embargo, algunas desventajas. En particular, para usarlos debemos suponer que los puntos espaciales donde se registran las series de tiempo no están correlacionados. Es decir, cada serie de tiempo es independiente de la otra. Esto, claramente no es razonable, sobre todo cuando los puntos están geograficamente muy cercanos entre ellos. Por otro lado, se puede considerar que todas las series están correlacionadas teniendo así una sola

[3] Este tipo de archivos puede leerse en Python cargando "from scipy.io import netcdf". También "import netCDF4".
[4] Ver el libro [8] en la carpeta de referencias en Dropbox con algunos ejemplo de uso

serie multivariada con una dimensión igual al número de puntos de la malla. Si consideramos toda la malla, tendríamos una serie multivariada de dimensión 6355 y longitud 26665, lo que la hace inmaejable en computadores personales. Justamente, la idea de este proyecto es hacer pronósticos de variables climáticas en todo el país usando herramientas prácticas/parsimoniosas que puedan ejecutarse en cualquier computador personal. Para lograr esto, se requiere disminuir la dimensionalidad del problema y al mismo tiempo considerar las posibles correlaciones entre las series. En resumen, el enfoque buscado debe facilitar la transición de un problema de predicción de datos espaciotemporales multivariados de alta dimensión a un problema de predicción de series temporales de baja dimensión. Esta transición buscará reducir sustancialmente la complejidad computacional, al mismo tiempo que pueda producir predicciones razonablemente precisas y permita mejorar nuestra capacidad para interpretar y predecir patrones climáticos en todo el territorio en horizontes temporales de mediano plazo. Todo esto, a pesar de la alta variabilidad climática del territorio.

1. Estrategia 1 para abordar el problema: Uso de Autoencoder (AE)

La estrategia propuesta aquí se basa en el uso de un *AE* para procesar series temporales de precipitaciones [2]. Específicamente, se considera un conjunto de series temporales que representan los registros de precipitaciones en q estaciones pluviométricas distribuidas en un territorio definido. Se estructurarán los datos de modo que cada columna sea una estación y cada fila una observación diaria de la precipitación caída, medida en milímetros (mm) (ver tabla 1). Con una parte de los datos se entrenará el *AE*, dejando reservado una parte de ellos para la validación de las predicciones (por ejemplo, usar el último año 2022 de los datos base para validar la predicción del modelo) .

Usando un *AE*, se logra disminuir la dimensión, al transformar la serie multivariada original $\bar{R}(t) = \left\{R_{t_i}^{r_j}\right\}$ con i =1 ...n, j = 1 ... q a vectores en un espacio latente $\bar{z}(t) = \left\{z_{t_i}^{l}\right\}$ con i =1 ... n, $l = 1$ ... dim_hidden empleando el codificador del *AE*. El espacio latente tiene una dimensión reducida (este proceso se le denomina "embedding"). Para cada tiempo t_i se tiene un vector $\boldsymbol{z}_{t_i}$ de dimensión dim_hidden igual a la dimensión del espacio latente[5].

Los vectores latentes constituyen una representación compacta de la serie multivariada original de alta dimensionalidad, que recogen información temporal y espacial relevante, disminuyendo en ruido e información superflua. En vez de analizar las series de tiempo en el espacio original directamente, la idea es tratar con los vectores latentes en el espacio oculto del *AE*. Es decir, hacemos el pronóstico sobre la serie temporal de vectores latentes $\bar{z}(t)$ a un horizonte h, $\bar{z}(t+h) = \left(\boldsymbol{z}_{t_1}, \boldsymbol{z}_{t_2}, \dots, \boldsymbol{z}_{t_j}, \dots \boldsymbol{z}_{t_n}, \overbrace{\boldsymbol{z}_{t_{n+1}} \dots \boldsymbol{z}_{t_n+h}}\right)$. Posteriormente, con ayuda del decodificador del

[5] El valor de esta dimensión se elige como hiperparámetro durante el entrenamiento del *AE* por ejemplo, se sugiere usar $dim\_hidden = 100$ en este problema.

AE, hacemos la transformación inversa, regresando al espacio original, pero con las series extendidas en un horizonte h en cada localización r_j. Es decir, ahora tenemos $\bar{R}(t+h) = \left\{R_{t_i}^{r_j}\right\}$ con i =1,2, ...n, $n+1, \ldots, n+h$ y $j = 1 \ldots q$

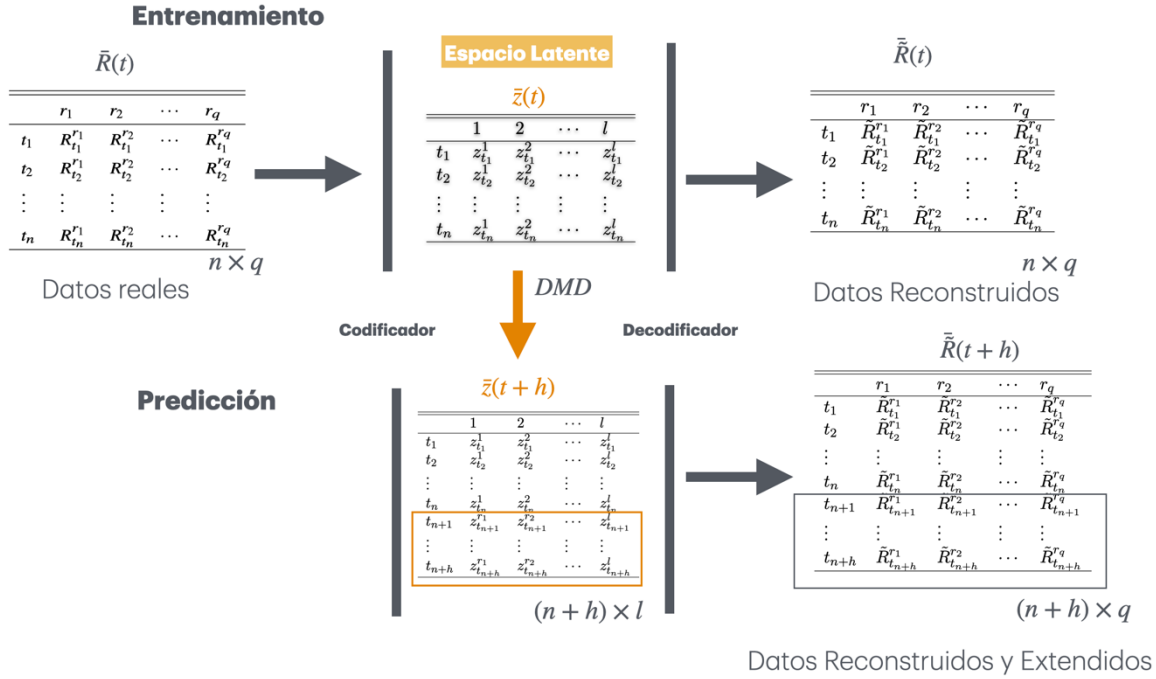

Figura 1: Representación esquemática del uso combinado del Autoencoder y Dynamic Mode Decomposition (DMD) para el pronóstico de series de tiempo.

Para la predicción en el espacio latente del *AE*, se empleará la Descomposición en Modos Dinámicos (DMD). Una vez entrenado DMD, permite realizar predicciones en el espacio latente, que luego pueden ser decodificadas por el *AE* para reconstruir la serie temporal, extendiéndola por h días hacia el futuro. Para esta tarea, se utilizará la biblioteca de Python **PyDMD** [3,16].

La **Descomposición de Modos Dinámicos** (DMD) [10, 11] es una técnica libre de ecuaciones, que opera con instantáneas en el tiempo de las mediciones (es decir, con series de tiempo) aliviando la necesidad de contar con un conjunto de ecuaciones rectoras de la dinámica. DMD es un método poderoso, desarrollado en la comunidad de dinámica de fluidos, con la capacidad de encontrar patrones espacio-temporales coherentes en los datos que surgen de sistemas no lineales de gran escala y dimensionalidad. En la práctica, se utilizará la biblioteca de Python PyDMD que permite un análisis cómodo usando esta técnica. Así, el objetivo es tomar las series de tiempo asociadas a los vectores latentes y usar DMD para descomponer la evolución de estos vectores en los denominados modos dinámicos. En DMD cada modo se representa como un vector con componentes complejas que es asociado a cada una de las componentes ($l = 1 \ldots dim\_hidden$) de los vectores latentes. Con ayuda del decodificador del *AE* podemos enviar cada modo dinámico al espacio real e interpretar correctamente el significado de estos modos. Una vez aplicado el decodificador se

puede interpretar el módulo y la fase de cada modo en el espacio real. Cada modo captura un patrón espacio-temporal de las precipitaciones en el territorio. Por ejemplo, un patrón de escases o por el contrario de lluvia intensa en ciertas zonas. El módulo del modo indica su intensidad y la parte imaginaria permite hallar la fase relativa (relativa significa que pueden haber desfases entre los modos dinámicos en dependencia del lugar, es decir de la ubicación de la estación pluviométrica, también pueden haber sincronizaciones de la dinámica entre varias estaciones). Al igual que en el análisis de componentes principales, en el análisis con DMD se halla un conjunto reducido de modos dinámicos que capturan casi toda la dinámica de las precipitaciones (los modos con más "energía") que se distribuyen en el territorio. Los modos dinámicos son vectores propios de una matriz que modela/aproxima de manera lineal la evolución de las series de tiempo, en este caso de la serie de tiempo de vectores latentes. Los valores propios de esta matriz, por otro lado, indican cómo evolucionan los modos dinámicos. Si todos los valores propios, en valor absoluto, son menores a 1, entonces los modos decaen en el tiempo, si alguno es mayor que 1, este se vuelve dominante. Las fases asociadas a la parte imaginaria de los valores propios indican procesos periódicos o repetitivos en la evolución. Estos valores pueden visualizarse en un mapa de colores y mostrar las diferencias entre la evolución de los modos dinámicos para las distintas localizaciones (estaciones pluviométricas) para obtener así una mejor interpretación de los patrones hallados. La biblioteca PyDMD cuenta con algunos recursos interesantes para la visualización de los modos, lo que permite una mejor interpretación. Pero los mapas deben construirse con otras herramientas. PyDMD cuenta además con opciones para incorporar parámetros en forma de control en el análisis de las series, lo que da la posibilidad de incluir el efecto de ciertos eventos como parámetros que afectan la dinámica. Por ejemplo eventos asociados a las oscilaciones del Niño-Niña[6] (ahora definidos como parámetros de control) en la dinámica de las precipitaciones.

2. Estrategia 2 para abordar el problema: Uso de Variational Autoencoder (VAE) con el operador de Koopman.

El autoencoder Variacional (VAE) [17] es un autoencoder que incorpora un termino de regulación en la función de coste de la red neuronal basado en la divergencia de Kullback – Leibler [14]. A diferencia del *AE* no establece una relación 1-1 del dato de entrada con su representación en el espacio latente, sino que el dato de entrada se pone en correspondencia con una distribución de probabilidad. El muestreo de esta serie en el espacio latente (habitualmente una distribución Gaussina) puede generar datos similiares al original. Es por eso que el VAE entra dentro de la categoría de IA generativa. La idea dada en [13] es incorporar otro término adicional en la función de coste basado en el operador de Koopman [16] , creandose el

[6] Datos sobre métricas asociadas a estos fenómenos pueden descargarse desde el sitio: https://origin.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/ONI_v5.php

KoVAE. El operador de Koopman es un operador lineal que permite linealizar la dinámica, y cuya aproximación más utilizada es precisamente el DMD, el mismo que comentamos en la estrategia anterior. La idea aquí es que muestreando la función de distribución en el espacio lantente del *KoVAE* se podría eventualmente obtener el pronóstico de las series de tiempo de manera similar a lo hecho en la primera estrategia, pero de manena más directa, dado que este VAE ya incorpora el DMD en durante el entrenamiento. Es decir, en la anterior estrategia, primero se entrena el AE y después de entrena, de manera independiente, el DMD (ver figura 1). La idea aquí es aplicar el *KoVAE* para la predicción masiva de series de tiempo de precipitaciones en todo el territorio de Chile. Algo que no se ha hecho aún. El código del preprint [13] sobre el *KoVAE* esta disponible en [15].

Link a carpeta Dropbox con información y algunas referencias mencionadas en el proyecto [19]

Referencias

[1] https://github.com/awslabs/gluonts
[2] https://github.com/mauricio-herrera/Spatiotemporal-Forecasting-Autoencoder-DMD
[3] https://github.com/PyDMD/PyDMD
[4] Wong, C., 2023: Deepmind AI accurately forecasts weather - on a desktop computer. Nature, https://doi.org/10.1038/d41586-023-03552-y.
[5] Lam, R., et.al. 2023: Learning skillful medium-range global weather forecasting. Science, 382 (6677), 1416–1421, https://doi.org/10.1126/science.adi2336
[6] Cressie, N., and C. Wikle, 2011: Statistics for Spatiotemporal Data. Wiley, Hoboken.
[7] Hersbach, H. et.al. ERA5 hourly data on single levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS) doi: 10.24381/cds.adbb2d47.
[8] Vitor Cerqueira, Luís Roque. Deep Learning for Time Series Cookbook, 2024 Packt Publishing.
[9] https://ts.gluon.ai/stable/
[10] J. Nathan Kutz, et. al. Dynamic Mode Decomposition Data-Driven Modeling of Complex Systems. 2016, Society for Industrial and Applied Mathematics
[11] Peter, Schmid. J. Fluid Mech. (2010), vol. 656, pp. 5–28. doi:10.1017/S0022112010001217, Cambridge University Press 2010.
[12] https://github.com/azencot-group/KoVAE
[13] Ilan Naiman, et. al. Generative Modeling of Regular and Irregular Time Series Data via Koopman {VAE}s, The Twelfth International Conference on Learning Representations. 2024, https://openreview.net/forum?id=eY7sLb0dVF.
[14] Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat*. 1951, 22, 79–86.
[15] https://github.com/azencot-group/KoVAE

[16] Bethany Lusch et. al. Deep learning for universal linear embeddings of nonlinear dynamics. NATURE COMMUNICATIONS | (2018) 9:4950 | DOI: 10.1038/s41467-018-07210-0 | www.nature.com/naturecommunications
[16] Sara M. Ichinaga, et. al. PyDMD: A Python package for robust dynamic mode decomposition
[17] Carl Doersch. Tutorial on Variational Autoencoders. arXiv:1606.05908v3, 2021
[18] Link a los datos base en carpeta de Dropbox
https://www.dropbox.com/scl/fo/y1sya6na9bmp0qzpzhkh1/ABAlCelkXVpNvjqz2EEfjb4?rlkey=i3wdm49qdfymuitks2s4kiuqp&dl=0
[19] Link a carpeta Dropbox con información y algunas referencias mencionadas en el proyecto:
https://www.dropbox.com/scl/fo/lsvcxiyz3b54elid95xft/ACCIPdIIru9pCJwXq3m5ji0?rlkey=flnernugznfjbvbcajwoupqvw&dl=0