Estructuras Modulares en Redes Complejas: Trabajo Práctico

Mikel Berganza Muguruza

13 de abril de 2021

Resumen

Esta documentación describe el desarrollo realizado para el trabajo final de la asignatura de estructuras modulares en redes complejas, en el que se analiza una red social y se plantean una serie de modelos de predicción a partir de unas características extraídas de la red social en cuestión. El trabajo se ha basado en el artículo [1].

Esta memoria consta de una primera sección donde se analiza la base de datos escogida para realizar el trabajo. En la segunda sección se pueden ver las medidas de centralidad utilizadas para la fase de análisis de la red social. En la tercera sección se pueden observar los clasificadores escogidos, y por último, se discuten los resultados obtenidos en la sección de resultados.

1. Base de datos

Siendo el objetivo principal de esta práctica la de analizar una red social, como punto inicial se ha escogido una base de datos de una red social. En este caso, se ha escogido la base de datos de *Epinions* de *Kaggle* (`https://www.kaggle.com/masoud3/epinions-trust-network`).

Ésta es una red social de confianza (quién confía en quién) en línea de un antiguo sitio de reseñas de consumidores generales (*Epinions.com*). Los miembros del sitio pueden decidir si "confiar" entre sí. Todas las relaciones de confianza interactúan y forman la red de confianza mencionada.

La base de datos de *Epinions* contiene las calificaciones de los usuarios sobre los elementos y las relaciones explícitas de confianza/desconfianza entre los usuarios. Para este proyecto, solo se van a utilizar las relaciones de confianza/desconfianza, y se representan de la siguiente manera (Tabla 1):

ID A	ID B	Relación	Fecha

Tabla 1: Estructura de los datos.

- **ID A.** La identificación del miembro que está haciendo la declaración de confianza/desconfianza.
- **ID B.** La segunda identificación es la identificación del miembro en quien se confía/desconfía.
- **Relación.** Valor de la relación, 1 si el usuario A confía en el B y -1 si desconfía.
- **Fecha.** La fecha en la que se hizo la relación de confianza/desconfianza.

Si bien la base de datos contiene cuatro columnas, la fecha de cuando se creó la relación de confianza/-desconfianza no es de utilidad en este caso, por lo que se va a descartar desde un comienzo.

2. Medidas de centralidad

Para realizar la fase de análisis de la red social, se han escogido las siguientes medidas de centralidad:

- **Centralidad de grado (*degree centrality*).** La centralidad de grado es la medida de centralidad más simple de calcular.El grado de un nodo es simplemente un recuento de cuántas conexiones sociales (es decir, arcos) tiene, en este caso, siendo un grafo dirigido, el grado total de un nodo es la suma de los arcos salientes y entrantes. Dos criterios para normalizar esta medida pueden ser dividir el grado de cada nodo por el máximo grado obtenido de la red, o bien dividirlo por el número total de nodos de la red.
- **Centralidad de grado de entrada (*indegree centrality*).** Este caso es parecido al anterior, sin embargo, solo se cuentan los arcos entrantes al nodo (número de vínculos o arcos dirigidos al nodo). En el sentido de relaciones interpersonales, puede interpretarse como una medida de popularidad.
- **Centralidad de grado de salida (*outdegree centrality*).** Inversamente al caso anterior, solo se cuentan los arcos salientes del nodo (número de vínculos que el nodo redirige a otros). En el sentido de relaciones interpersonales, puede interpretarse como una medida de sociabilidad.
- **Centralidad alfa (alpha centrality).** La centralidad alfa es una variante de la centralidad de vector propio en que los nodos están sujetos a distinta importancia dependiendo de factores externos. Un proceso intuitivo para calcular la centralidad de los vectores propios (*eigenvectors*) es dar a cada nodo una cantidad inicial de influencia positiva aleatoria. Luego, cada nodo divide su influencia de manera uniforme y la divide entre sus vecinos externos, recibiendo de sus vecinos internos en especie. Este proceso se repite hasta que todos están dando tanto como ingiriendo y el sistema ha alcanzado un estado estable. La cantidad de influencia que tienen en este estado estable es su centralidad de vector propio. La centralidad alfa mejora este proceso al permitir que los nodos tengan fuentes externas de influencia (la cantidad de influencia que recibe el nodo i en cada ronda está codificada en e_i) [2].
- **Centralidad de poder (power centrality).** La centralidad de poder es una generalización de la centralidad de grado. En esta centralidad, para un nodo se cuenta el número de nodos que están conectados con él a través de un camino (no tienen que ser de distancia 1 como en el caso de la centralidad de grado), al mismo tiempo que se penalizan las conexiones con nodos más distantes por medio de un factor β, el cual puede ser negativo [3].
- **Centralidad de intermediación (betweenness centrality).** La centralidad de intermediación es una medida de centralidad en un gráfico basada en las rutas más cortas. Es una medida que cuantifica la frecuencia o el número de veces que un nodo actúa como un puente a lo largo del camino más corto entre otros dos nodos [4].
- **Centralidad de subgrafo (subgraph centrality).** La centralidad de subgrafos de un vértice mide el número de subgrafos en los que participa un vértice, ponderándolos según su tamaño. Se define como el número de bucles cerrados que se originan en el vértice, donde los bucles más largos tienen una ponderación exponencialmente reducida [5].

Las medidas de centralidad no han sido escogidas con ningún criterio en específico. Pero cabe destacar que han sido las únicas medidas de centralidad que no han dado errores de computo, y que funcionaban con normalidad, puesto que la mayoría de centralidades daban errores de tamaño de vector en el lenguaje de R utilizado, aún reduciendo el número de nodos del grafo a valores como 50. Por ello, se han descartado muchas medidas de centralidad y se han dejado únicamente las que funcionaban.

Las centralidades anteriormente mencionadas calculan el nivel de centralidad de los vértices en el grafo. Por ello, a la hora de calcular las centralidades de grafo, se ha utilizado el método de centralización (función

centralize en R). El cual, es un método para crear una medida de centralización a nivel de gráfico a partir de las puntuaciones de centralidad de los vértices.

La formula para obtener la centralización a nivel de grafo es la siguiente:

$$C(G) = sum(max(c(w), w) - c(v), v),$$

donde $c(v)$ es la centralidad del vértice v.

Son estas medidas de centralidad las que se han utilizado a la hora de entrenar y testear los modelos de clasificación.

Queda comentar que como en un principio se escogieron una misma cantidad de enlaces para cada clase (confianza y desconfianza), no se ha visto la necesidad de usar la técnica de SMOTE para balancear las clases.

3. Clasificadores

Los modelos de clasificación escogidos han sido los siguientes:

- **k vecinos más cercanos (*k-nn*).** El algoritmo k-nn es uno de los algoritmos de clasificación más simples. La idea básica sobre la que se fundamenta este modelo es que un nuevo caso se va a clasificar como la clase más frecuente a la que pertenecen sus K vecinos más cercanos (generalmente usando la distancia euclidiana).

- **Clasificador bayesiano ingenuo (*Naive Bayes*).** Los clasificadores ingenuos de Bayes son una familia de "clasificadores probabilísticos" simples basados en la aplicación del teorema de Bayes con fuertes (ingenuas) suposiciones de independencia entre las características.

  Todos los clasificadores de Bayes ingenuos asumen que el valor de una característica en particular es independiente del valor de cualquier otra característica, dada la variable de clase. Por ejemplo, una fruta puede considerarse una manzana si es roja, redonda y de unos 10 cm de diámetro. Un clasificador de Bayes ingenuo considera que cada una de estas características contribuye de forma independiente a la probabilidad de que esta fruta sea una manzana, independientemente de las posibles correlaciones entre las características de color, redondez y diámetro.

- **Bosques aleatorios (*Random Forest*).** El bosque aleatorio es una combinación de árboles predictores en el que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Básicamente, se basa en construir una multitud de árboles de decisión en el momento del entrenamiento y generar la clase que es el moda de las clases (en clasificación) de los árboles individuales [6].

- **Árbol de decisión (*Decision Tree*).** Un árbol de decisión es un modelo con forma de árbol, el cual está basado en reglas o condiciones para separar los datos. Los árboles de decisión se construyen utilizando una heurística llamada partición recursiva. Este enfoque también se conoce comúnmente como dividir y conquistar porque divide los datos (en la característica que da como resultado la mayor ganancia de información) en subconjuntos, que luego se dividen repetidamente en subconjuntos aún más pequeños, y así sucesivamente hasta que el proceso se detiene cuando el algoritmo determina que los datos dentro de los subconjuntos son suficientemente homogéneos, o se ha cumplido otro criterio de detención.

- **Red neuronal (*Neural network*).** Una red neuronal consta de muchas capas diferentes de neuronas, y cada capa recibe entradas (*inputs*) de capas anteriores y pasa salidas (*outputs*) a las siguientes. La

forma en que la salida de cada capa se convierte en la entrada para la siguiente capa depende del peso que se le dé a ese enlace específico, que depende de la función de costo y del optimizador.

La red neuronal itera durante un número predeterminado de iteraciones, llamadas épocas (del ingles, *epochs*, es cuando un conjunto de datos completo se pasa hacia adelante y hacia atrás a través de la red neuronal una única vez). Después de cada época, se analiza la función de costos para ver dónde se podría mejorar el modelo.

Por último, la función de optimización altera la mecánica interna de la red, como los pesos y los sesgos (*bias*), en base a la información proporcionada por la función de costo, hasta que la función de costo sea minimizada.

Para la fase de entrenamiento y testeo, se ha escogido el método de *10-fold cross-validation* (validación cruzada) como método de remuestreo (*resampling*). Para que los resultados obtenidos de cada modelo pudieran ser comparables, se han mantenido en todo momento las particiones iguales de la validación cruzada.

4. Resultados

Las medidas de rendimiento escogidas para evaluar los modelos han sido: exactitud (*accuracy*), precisión, exhaustividad (*recall*), valor-F y valor MCC (*Matthews correlation coefficient*).

En la Tabla 4 se pueden observar los resultados obtenidos para los distintos modelos.

	Accuracy	Precision	Recall	F_Score	Mcc_Score
KNN	0.98	0.98	0.98	0.98	0.96
NB	0.67	0.60	1.00	0.75	0.45
RF	1.00	1.00	1.00	1.00	1.00
DT	1.00	1.00	1.00	1.00	1.00
NN	0.68	1.00	0.36	0.53	0.47

Tabla 4: Tabla de resultados de los modelos.

Para estudiar la significatividad de los datos, se han realizado dos pruebas. La primera de ellas ha sido un test de normalidad sobre los vectores de resultados del *accuracy* (exactitud) de cada clasificador. Para ello, se ha utilizado el test de Shapiro–Wilk, el cual se usa para contrastar la normalidad de un conjunto de datos.

El test de Shapiro-Wilk se interpreta de la siguiente manera: siendo la hipótesis nula que la población está distribuida normalmente, si el p-valor es menor o igual a alfa (nivel de significancia, normalmente 0.05) entonces la hipótesis nula es rechazada (se concluye que los datos no vienen de una distribución normal). Si el p-valor es mayor a alfa, se concluye que no se puede rechazar dicha hipótesis.

Como se puede ver en el siguiente fragmento de "código" en *R*, todos los modelos han dado un p-valor menor a alfa. Por ello, se concluye que no se puede asumir normalidad en el conjunto de datos.

R code 4.1: Resultados de los test de Shapiro-Wilk de los distintos modelos.

```
> shapiro.test(KNNTrain[["resample"]][["Accuracy"]])

        Shapiro-Wilk normality test

data:  KNNTrain[["resample"]][["Accuracy"]]
W = 0.50927, p-value = 4.672e-06

```

```
>
> shapiro.test(NBTrain[["resample"]][["Accuracy"]])

        Shapiro-Wilk normality test

data:  NBTrain[["resample"]][["Accuracy"]]
W = 0.82474, p-value = 0.02892

>
> shapiro.test(RFTrain[["resample"]][["Accuracy"]])

        Shapiro-Wilk normality test

data:  RFTrain[["resample"]][["Accuracy"]]
W = 0.36572, p-value = 1.004e-07

>
> shapiro.test(DTTrain[["resample"]][["Accuracy"]])

        Shapiro-Wilk normality test

data:  DTTrain[["resample"]][["Accuracy"]]
W = 0.36572, p-value = 1.004e-07

>
> shapiro.test(NNTrain[["resample"]][["Accuracy"]])

        Shapiro-Wilk normality test

data:  NNTrain[["resample"]][["Accuracy"]]
W = 0.79895, p-value = 0.01408
```

Como no se ha podido asumir la normalidad en los datos, se ha realizado posteriormente el test de Kruskal-Wallis.

Es el test adecuado cuando los datos tienen un orden natural, es decir, cuando para darles sentido tienen que estar ordenados o bien cuando no se satisfacen las condiciones para poder aplicar un ANOVA, como puede ser la condición de normalidad de los datos. Por ello, se ha realizado el test de Kruskal-Wallis.

El test de Kruskal-Wallis contrasta si las diferentes muestras están equidistribuidas y que por lo tanto pertenecen a una misma distribución (población). En este caso, se toma como hipótesis nula que todas las muestras provienen de la misma población (distribución).

En el siguiente fragmento, se puede observar el resultado obtenido al realizar el test de Kruskal-Wallis sobre la población. Hay que denotar que se han cambiado los nombres a la hora de realizar la llamada para facilitar el entendimiento, en el código real tienen diferentes nombres, pero los resultados son los mismos.

R code 4.2: Resultados del test de Kruskal-Wallis.

```
> kruskal.test(list(KNNTrain[["resample"]][["Accuracy"]],
+                   NBTrain[["resample"]][["Accuracy"]],
+                   RFTrain[["resample"]][["Accuracy"]],
+                   DTTrain[["resample"]][["Accuracy"]],
+                   NNTrain[["resample"]][["Accuracy"]])

        Kruskal-Wallis rank sum test
```

```
8
9  data:  list(KNNTrain[["resample"]][["Accuracy"]],
       NBTrain[["resample"]][["Accuracy"]],
10 RFTrain[["resample"]][["Accuracy"]], DTTrain[["resample"]][["Accuracy"]],
       NNTrain[["resample"]][["Accuracy"]])
11 Kruskal-Wallis chi-squared = 37.453, df = 4, p-value = 1.453e-07
```

Como se puede ver en el fragmento de "código" de R, se obtiene un p-valor mucho menor a alfa (0.05). Por ello, se rechaza la idea de que la diferencia se debe al muestreo aleatorio y, en cambio, se puede concluir que las poblaciones tienen distribuciones diferentes.

Por lo tanto, es significativo el hecho de que un modelo sea mejor que otros basado en las medidas de rendimiento obtenidas. En este caso, y viendo los resultados de la Tabla 4, se puede concluir que los modelos de *Random Forest* y *Decision Tree* han sido los mejores clasificadores.

Referencias

[1] Víctor Hugo Masías, Mauricio Valle, Carlo Morselli, Fernando Crespo, Augusto Vargas Schüler, and Sigifredo Laengle. Modeling verdict outcomes using social network measures: The watergate and caviar network cases. *PloS one*, 11:e0147248, 01 2016.

[2] Phillip Bonacich and Paulette Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social networks*, 23(3):191–201, 2001.

[3] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 03 1987.

[4] J. Sun and Jie Tang. *A Survey of Models and Algorithms for Social Influence Analysis*, volume 1, pages 177–214. Springer, 03 2011.

[5] Ernesto Estrada and Juan A Rodriguez-Velazquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.

[6] S Madeh Piryonesi and Tamer E El-Diraby. Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. *Journal of Transportation Engineering, Part B: Pavements*, 146(2):04020022, 2020.