1.5em 0pt

Estructuras modulares en redes complejas

Maider Ibarra

April 2021

Abstract

Debido al auge del blockchain y del anonimato de los usuarios que usan las criptomonedas surgen vías alternativas para evitar transacciones fraudulentas. Una de ellas es la valoración de los usuarios teniendo en cuenta sus interacciones. En este artículo se utilizan diferentes algoritmos de clasificación en una base datos que almacena una who-trusts-whom network de consumidores de criptomonedas. El objetivo del documento es realizar una investigación para determinar que algoritmos son los más adecuados para porder clasificar las transacciones en fraudulentas y no fraudulentas.

1 Introduction

El artículo consta de los siguientes apartados, para documentar el proceso seguido en la documentación:

1. Estado del arte
2. Descripción datos
3. Preprocesamiento
4. Clasificación
5. Experimentación
6. Conclusiones

2 Estado del arte

Las criptomonedas surgieron el 3 de enero de 2009, fueron creadas como una divisa digital. Mediante ellas, el trueque o intercambio se hace digitalmente. Entre sus ventajas se encuentran, la posibilidad de realizar operaciones instantaneamente y desde cualquier parte del mundo. Una de las características de la criptomoneda es que no es controlada ni respaldada por ninguna entidad por lo que este tipo de moneda no es rasteable.

3 Descripción datos

El dataset utilizado para realizar esta investigación 'Bitcoin Alpha trust weighted signed network' [2] [1] pertenece a la universidad de Stanford. En ella se almacenan valoraciones entre usuarios, con el objetivo de identificar usuarios fiables y no fiables de la página web https://btc-alpha.com/es/.

Este portal es la plataforma Europea de cambio de criptomoneda, donde se realizan transacciones de forma segura y fiable.

Para garantizar la fiabilidad los usuarios realizan valoraciones de otros usuarios, puntuando su fiabilidad en un rango de $[-10, 10]$. El análisis de estos datos es de vital importancia para btc-alpha. Ya que, los consumidores de criptomonedas son anónimos por lo que llevar un traceo de la reputación de los usuarios es la única manera de evitar fraudes. El dataset utilizado es la primera red direcionada con pesos libre para investigación.

Las estadísticas del dataset son los siguientes:

- Nodes: 3,783 Usuarios de la alicación
- Edges: 24,186 Puntuaciones.
- Range of edge weight: $[-10, +10]$
- Percentage of positive edges: 93%

Los campos del dataset son los siguientes:

- Source: id del nodo de origen, quien realiza la puntuación.
- Rating: Puntuación $[-10, +10]$.
- Target: id del nodo de destino, quien recibe la puntuacio.
- Time: Fecha en la que se realiza la puntuación

4 Preprocesamiento

Antes de empezar con la clasificación se ha amoldado el dataset para poder procesarlo. Los pasos seguidos han sido los siguientes:

- Resampling del dataset dejando 500 nodos.
- Cambiar valores Na por 0.
- Añadir el campo class, 0 cuando el rating es negativo y 1 cuando es positivo.
- El dataset esta umbalased, es decir, tiene más transacciones con rating positivo que negativo por lo que se utiliza la técnica SMOTE. Es una técnica de oversampling, la cual crea nuevos puntos ficticeos teniendo en cuenta los k nearest neighbors según los parámetros y un factor aleatorio.

Elige el vector entre un punto y uno de sus k nearest neightbors y lo multiplica por un factor aleatorio, surgiendo un nuevo punto ficticio. De esta manera se evitan los problemas que causa tener las clases desbalanceadas como por ejemplo, el acuraccy paradox. En este caso como el 93% de las puntuaciones son positivas, los algoritmos que el 100% obtengan como resultado de la clasificación positivo tendrían muy buena accuracy a pesar de no ser buenos algoritmos.

- Se añaden nuevas columnas, las cuales corresponden a medidas de centralidad de la red, para posteriormente poder clasificar las transacciones teniendo en cuenta la estructura de cada nodo. Se utilizan las siguientes medidas de centralidad:

  - Degree centrality: Es la cantidad o número de conexiones vinculadas a un vértice. En las redes dirigidas, como la que nos ocupa en éste ejercicio (las relaciones tienen un origen y un destino en lugar de conexiones mutuas, es decir los lazos tienen dirección); existen dos versiones de la medida de grado, centralidad de entrada y centralidad de salida.
  - In-degree centrality: La centralidad de entrada, se corresponde con el número de conexiones dirigidas al vértice, que apuntan hacia adentro de un vértice (puntos finales de la cabeza). Es el número de enlaces que entran.
  - Out-degree centrality: La centralidad de salida, se corresponde con el número de conexiones que desde un vértice se dirigen y apuntan a otros vértices (puntos finales de cola). De otra manera es el número de enlaces salientes o el número de vértices sucesores.
  - Eigen vector centrality: Mide la importancia de un nodo-vértice, en relación a sus vecinos. Por ello se puede utilizar para determinar la influencia en las redes sociales.
    Su fundamento se base en que los enlaces de nodos-vértices con alto grado de centralidad tienen más valor que los enlaces con bajo grado de centralidad. Es decir los nodos con más bordes ganan en importancia, y esta importancia se traslada a los nodos a los que están conectados.
  - Betweenness centrality: Mide el número de veces que un nodo-vértice se halla en el camino entre otros nodos-vértices aleatorios.
  - Constraint: El grado de resticción de un nodo por su red.
  - Effective network size: Mide la falta de rebundancia de una red.
  - Clustering coefficient: Mide el grado en que los vértices de un gráfo tienden a juntarse entre sí. Si el vértice está agrupado formando un "grafo completo" su valor es máximo.

5 Clasificación

Dado que el objetivo de este artículo es determinar que algoritmo es el mejor para predecir el grado de confianza que deposita un usuario de la página web btc-alpha en otro, se han estudiado los siguientes algoritmos:Random forest, Naive Bayes, KNN, Association rules y Neural network.

A la hora de particionar el dataset en trainning y testing se ha utilizado la técnica 10 fold cross validation. De este mpdo, el data set se ha dividido en 10 partes del mismo tamaño y el algoritmo se ha ejecutado en un bucle de 10 iteraciones. En cada iteración se ha utilizado una parte k para la validación y las demás partes para entrenar el algoritmo. Cada una de las iteraciones ha utilizado una parte diferente para el testeo. Es decir, ninguna parte se ha utilizado como validación más de una vez.

Las métricas utilizadas para valorar los resultados del testing-set de los modelos creados, han sido las siguientes:

- Accuracy: La métrica accuracy, mide el porcentaje de acierto del modelo, indicando la cercanía del resultado de una medición al valor verdadero. Dicho de otro modo, indica el número de elementos clasificados correctamente en comparación con el número total, aportando la cantidad de predicciones positivas que fueron correctas.

  $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$

- Precision: Se utiliza para medir la calidad del modelo de machine learning. Cuánto menor es la dispersión de los valores obtenidos, mayor será la precisión. Esta métrica, representa la proporción entre el número de predicciones correctas (tanto positivas como negativas) del algoritmo y el total de predicciones.

  TP: True Positive (Verdaderos Positivos)
  FP: False Positive (Falsos positivos)

  $Precision = \frac{TP}{TP+FP}$

- Recall: Se relaciona con la capacidad que tiene el modelo de machine learning para identificar la clase.
  TP: True Positive (Verdaderos Positivos)
  FP: False Positive (Falsos positivos)

  $Recall = \frac{TP}{TP+FN}$

- MCC: Se utiliza para clasificación binaria. Cuando el clasificador es perfecto (FP = FN = 0), el valor de MCC es 1, lo que indica una correlación positiva perfecta. Por el contrario, cuando el clasificador siempre clasifica erróneamente (TP = TN = 0), obtenemos un valor de -1, que representa una correlación negativa perfecta.

$MCC = \frac{TPXTN - FPXFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

- F-Score: El valor F1 compara el rendimiento de las métricas precisión y recall.

  $FScore = 2x\frac{precisiónxrecall}{precisión+recall}$

5.1 Random forest

Random forest utiliza un conjunto de algoritmos decision trees, los cuales realizan sus clasificaciones por separado, siendo el resultado del random forest la clase más predecida por la mayoría de los decision trees. El punto fuerte de este algoritmo es que los arboles se protegen entre ellos de los errores individuales.

5.2 Naive Bayes

Naïve Bayes, es una técnica simple para construir clasificadores.Estos algoritmos asumen que los valores de cada característica son independientes a las demás características.

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

5.3 KNN

Cada muestra del dataset se situa en un espacio multidimensional, en el trainning las muestras son colocadas en el espacio y se les asigna su clase. A la hora de realizar la predición se situa el punto a predecir en el espacio, se buscan sus K vecinos más cercanos con una métrica de distancia (distancia euclidea, manhatan...) y se le asigna la etiqueta que tienen esos vecinos cercanos.

5.4 Neural network

Las redes neuronales son un conjunto de neuronas, las cuales tienen unos pesos que van cambiando, a medida que la red neuronal va aprendiendo ajustando el resultado (minimizando errores). Los pesos van dando más valor a unas neuronas que a otras. Finalmente se obtiene un resultado.

Las redes neuronales imitan el comportamiento del cerebro humano, y van cambaindo a medida que surgen nuevas investigaciones.

6 Experimentación

Los resultados de las métricas obtenidas aplicando los modelos han sido los siguientes:

Algoritmo	accuracy	recall	f1 score	MCC
Random forest	0.8051675977653632	0.8051675977653632	0.6760395567735017	0.1382350484446506
Naïve Bayes	0.5966014897579144	0.5966014897579144	0.6244853737811484	0.19536945690130017
KNN	0.8032472067039106	0.8032472067039106	0.8331770859031924	0.64976489267773
Association rules	0.7616387337057728	0.7616387337057728	0.8023356818839881	0.5742199583477231
Red neuronal	0.5253142458100558	0.5253142458100559	0.6760395567735017	0.1382350484446506

Para comprobar la normalidad de los resultados se utiliza shapiro wilk test, en el cual se obtiene, statistic=0.8153307437896729, pvalue=0.00041424078517593443. Dado que pvalue es mucho menor que α (0.05) se rechaza la hipótesis H_0 los datos no presentan una distribución normal.

Al no presentar una distribución normal se realiza el test de Kruskal Wallis, el cual da el siguiente resultado:statistic=2.8990881458966564, pvalue=0.4074469027286959. Al ser $p > 0,05$ no se tiene evidencia para rechazar la hipotesis de que las diferencias entre las medianas no son estadísticamente significativas.Por lo que, estadísticamente no se puede apreciar que un algoritmo es mucho mejor que otro.

A pesar de ello, en la tabla de resultados se observa que el algoritmo que mejores métricas ha obtenido es KNN.

References

[1] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 333–341. ACM, 2018.

[2] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 221–230. IEEE, 2016.