Avances del trabajo con Prof. Guillermo Restrepo

Andrés C. Marulanda*

Universidad de Antioquia
acamilo.marulanda@udea.edu.co

27 de octubre de 2020

Resumen

En este documento voy a escribir algunos detalles de los avances en el trabajo con el Profesor Guillermo Restrepo, así como preguntas e ideas que puedan surgir en el camino.

1. Introducción

El objetivo es obtener un algoritmo capaz de hacer predicciones utilizando datos de sustancias actuales (hasta cierta fecha). Para esto, el objetivo es leer el sistema periódico (PS) de la misma manera que lo vió Mendeleev para hacer sus predicciones; esto es, saber en qué partes la similaridad es vertical, en cuales es horizontal, diagonal, etc, y cómo hacer predicciones sobre nuevos compuestos con este conocimiento. Inicialmente pensamos en redes neuronales convolucionales que tomen como input datos representados en un esquema de tabla periódica, similar a lo que hicieron en [1].

Adicionalmente podemos intentar interpretar (disecar) la red neuronal. Esto lo podemos hacer de la misma manera que se hace para las redes que se ocupan de tareas de visión artificial, por lo que podemos ver cada uno de los filtros y como estos se comportan a medida que los inputs varían.

Esto podría dar la información de cómo leer algún PS, sea el de Mendeleev o cualquier otro, pues nos podría dar información de en qué partes del PS importan las similaridades verticales, diagonales, etc. El código desarrollado hasta este momento toma como entrada un custom PS, esto es, un PS escrito por el usuario. Este puede ser el de Mendeleev, el desarrollado por el grupo de Prof. Restrepo o cualquier otro. La utilidad de esto recae en que puede darnos alguna idea de cómo es la convergencia del algoritmo con diferentes PSs, y con esto evaluar el contenido de información de cada uno. Así podrían compararse algoritmos para leer el PS de Mendeleev contra PS organizados aleatoriamente, entre otros.

Hasta este punto se tendría un "Mendeleev robot", al cual podemos examinar para ver como toma decisiones.

Si logramos esto, digamos, para el PS (y sustancias) de 1868, podemos pensar en hacer lo mismo para varios años alrededor de éste y evaluar como era la capacidad predictiva del PS de este año. Esto podría dar alguna evidencia de que 1868 fue el año (o la época) más apropiada para la creación del PS. En este punto cabe aclarar que ya se tiene listo un artículo sobre este tema, donde se muestra que el PS estaba listo para ser formulado desde 1840.

Más adelante podrían entrenarse sistemas similares para no sólo predecir probabilidad de existencia, sino también para aproximar algunas propiedades de compuestos por reemplazamiento de un elemento. Esto podría ser de utilidad, por ejemplo, si se buscara predecir propiedades de los compuestos formados con elementos desconocidos hasta la fecha, que es precisamente lo que hizo Mendeleev con el estaño y otros.

Se quiere utilizar sustancias mas PS como entrada para un algoritmo. Digamos, por ejemplo, que para las sustancias $R-Na_n$ y $R-K_n$ existen propiedades medidas. Luego, el objetivo del algoritmo será predecir esta propiedad para, p.e., $R-Fe_n$ (Fe o cualquier otro). Esta propiedad puede ser alguna propiedad experimental, o probabilidad de existencia, etc.

El algoritmo se encarga en el primer caso de aproximar la función

*Corresponding author

$$f(R - x_n) = f(x, existentR - Y_n, PS) \quad (1)$$

Y en el segundo caso de calcular la siguiente probabilidad:

$$P(existeR - X | existeR - Y + PS) \quad (2)$$

Con X cualquier elemento conocido en la época, Y un elemento tal que R−Y_n existe en la época considerada y PS es el sistema periódico de esa época.

Del algoritmo se espera que evalúe la similaridad entre X y Y, de manera que si estos son similares, la probabilidad de que exista un compuesto resultado de la sustitución de estos elementos debería de ser alta, y sus propiedades similares. Esta información de similaridad debería de poder encontrarse, por supuesto, en el PS.

2. Tratamiento de datos

El input puede ser un esquema de la tabla periódica (PT), construido de la siguiente manera.

Suponga que existe un conjunto de elementos A = $\{A_i\}$ y un fragmento de sustancia R tal que R−A_n existe en la época considerada. Entonces en nuestra representación, en la posición de cada uno de los elementos de A en la PT se asignará un valor de 1. De otra manera, el valor será de 0. Ahora bien, la pregunta es: existe la sustancia R-Y, para un elemento conocido Y? Este elemento Y tendrá un valor de -1 en la representación de la PT (PTR). En caso que R-Y exista en esta época, el label (y) es igual a 1, de otra manera es igual a 0. Un ejemplo de esta PTR se da en la figura 1, en la que se indica que los elementos K, Mg, O, S comparten una composición R-X_n en común, esto es, el compuesto RX_n existe en el conjunto de datos con X = K, Mg, O, S. Adicionalmente se indica la pregunta: Existe el compuesto R-Br_n?

Similarmente, en el caso de la predicción de propiedades, estos valores se intercambian por los valores reales de las sustancias consideradas.

2.1. Generación de los datos

Se encuentra cada fragmento R_n en el dataset, luego se encuentra una lista de elementos A = A_i tal que R−A_n exista para cada n; lo que produce una cantidad N(R_n) de fragmentos R_n y por lo tanto de PTRs. El nuevo dataset contiene la información correspondiente a las similaridades entre elementos, así como los R_n con los que se combinan.

Figura 1: Representación de la tabla periódica (PTR). (Arriba/abajo) representación incluyendo/sin serie de lantánidos.

En este punto se hace un análisis exploratorio de los datos (EDA), para lo cual se pueden tener consideraciones que se exponen en la siguiente sección.

El tamaño del dataset para alimentar al algoritmo efectivamente es expandido al incluir la información sobre las sustancias que quieren predecirse, esto es, mediante un "labeling"del dataset. Para cada PTR puede sugerirse al algoritmo que ejecute la predicción de propiedades para X elementos, lo que incrementa el tamaño del dataset a N(R_n)*X.

2.2. Algoritmo

Suponga que tiene un compuesto R-X_n, con $n \geq 1$. Consideramos como un nuevo R′ la combinación RX, de manera que tenemos un “nuevo compuesto” RX-X_{n-1} = R′-X_{n-1}. Puede continuarse de esta manera hasta agotar n, esto es, pueden crearse n-1 nuevos R′ para un total de n Rs diferentes a partir de uno sólo de los elementos de uno de los compuestos. Con esto se genera una cantidad de R_ns igual a la suma de todos los subíndices de cada elemento en todos los compuestos.

Naturalmente esto aumenta la cantidad de datos disponibles, pues se explota cada uno de los subíndices de cada elemento en cada fórmula molecular, lo cual es una ventaja en términos computacionales pues puede llevar a la reducción del overfitting y otros problemas que se encuentran durante la producción de modelos en machine learning (ML).

Por ejemplo, las fórmulas $C_6BrCl_2H_3$ y $C_6Cl_3H_3$ pueden compararse así: pueden escribirse como $C_6Cl_2H_3-Br$ y $C_6Cl_2H_3-Cl$. Esto lleva a que Cl y Br comparten este ligando en común, mostrando una similaridad que no se expresa con métodos más simples.

3. Análisis de datos

Una vez se tengan las PTR, se procede con el análisis de los datos. Esta sección se divide en dos: análisis exploratorio (EDA), en el que se quiere obtener datos estadísticos y medidas generales del dataset, y modelación, donde se quiere intentar dar estructura, significado y utilidad a los datos generados.

3.1. EDA

Ah pues aquí se hacen cosas :|

3.2. Modelación

3.2.1. Clustering

En esta sección se pretende utilizar métodos de ML y los datos obtenidos para dar estructura, significado y utilidad a los datos generados. El primer método que puede usarse es clustering. En este se utilizan algoritmos para encontrar clusters de elementos (no estrictamente en el sentido químico) similares dada una representación. En este caso al aplicarlo a las PTR, buscamos las combinaciones de elementos que más ocurren en estas. Debe recordarse que cada PTR representa a una sustancia por medio de los elementos que la componen y son reemplazables entre sí, lo que sugiere similaridad. Si los elementos de uno conjunto son en verdad similares, esto se repetirá constantemente a través de todas las sustancias consideradas, por lo que los clusters con mayor número de PTRs corresponderán a los grupos de similaridad más evidentes. En general lo que se obtiene son clusters representado grupos de similaridad entre elementos, lo que en la visión del PS como hipergrafos ordenados se nombra "hyperedges".

3.2.2. Algoritmo predictor

La idea inicialmente es utilizar redes neuronales para aproximar la función descrita por las ecuaciones 1 y 2. La arquitectura a usar es uno de los grandes problemas debido a que no existe una única manera de obtener una óptima y usualmente la aproximación es una heurística, donde se sugiere cómo se quiere que el algoritmo lea los datos. Adicionalmente en las arquitecturas convencionales existen limitaciones en términos del tipo de datos que pueden usarse, por lo que esto debe pensarse detenidamente. Por ejemplo, las redes convolucionales (CNN) se usan para la lectura de imágenes; si se quisiera dar como input una imagen mas un vector, tendría que pensarse en algo más.

Un problema de nuestra representación, sin embargo, es que no conserva nada de la información del fragmento R ni del subíndice n, sino sólo de los elementos que se combinan con R en la proporción n. Esto puede resultar problemático sobre todo en la aplicación de estimación de propiedades de nuevos compuestos. Por ejemplo el compuesto C_6H_5-Cl es un clorobenceno, y claramente puede prepararse un análogo con Br y I. En este caso R es C_6H_5. Similarmente ocurre con NaCl, donde se pueden también formar NaBr y NaI. Claramente los Rs son muy diferentes, así como los compuestos formados por estos, de manera que la obtención de propiedades de compuestos únicamente a partir de la representación hasta aquí formulada probablemente deba ser replanteada.

Una solución puede ser convertir este R en un vector, donde cada entrada indica la cantidad de equivalentes de un elemento que se encuentran en este R, y su longitud es la cantidad de elementos únicos en el dataset. El problema de esto es que no es compatible con la arquitectura de una CNN.

Esto puede arreglarse creando una nueva arquitectura de manera que permita ambas representaciones simultáneamente. En esta arquitectura, se lee por un lado la representación en tabla periódica mediante capas convolucionales, y por el otro el vector R por medio de redes neuronales densas (DNN) convencionales. Estas dos arquitecturas generan cada una un vector, que puede concatenarse para generar una única entrada a una sóla red, que producirá la salida esperada, sea probabilidad de existencia o propiedades de compuestos, etc, similar a lo que hacen p.e. en [2]. Esta arquitectura es ilustrada en la figura 2.

Referencias

[1] X. Zheng, P. Zheng, and R.-Z. Zhang, "Machine learning material properties from the periodic table using convolutional neural networks," *Chem. Sci.*, vol. 9, pp. 8426–8432, 2018.

[2] W. Hong, N. Zheng, Z. Xiong, and Z. Hu, "A parallel deep neural network using reviews and item metadata for cross-domain recommendation," *IEEE Access*, vol. 8, pp. 41774–41783, 2020.

Figura 2: Arquitectura propuesta de red neuronal.