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

Después de unas 2 reuniones nos reunimos nuevamente para revisar el plan de trabajo del grupo de investigación (y de él mismo), y lo que yo haría. Decidimos entonces trabajar en un algoritmo capaz de hacer predicciones utilizando datos de sustancias actuales. Esto se puede partir en 2:

- A partir de unas sustancias generar un sistema periódico, y con este mas las sustancias que existen, predecir sustancias.
- La otra aproximación es usar directamente las sustancias para hacer predicciones.

Cualquiera de estas dos puede explorarse, cada cual con sus ventajas y dificultades. Sin embargo a nosotros nos interesa la primera, pues queremos intentar ver a la tabla periódica de la misma manera que la vió Mendeleev para hacer sus predicciones; esto es, saber en qué partes la similaridad es vertical, en cuales es horizontal, diagonal, etc.

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, si se tienen buenos resultados, 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 çustom PS", esto es, un PS seleccionado 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 de alguna manera 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 la tabla periódica (y sustancias) de 1868, podemos pensar en hacer lo mismo para varios años alrededor de 1868 y evaluar como era la capacidad predictiva de la tabla periódica de este año. Esto podría dar alguna evidencia de que 1868 fue el año (o la época) más apropiada para la creación del sistema periódico. En este punto cabe aclarar que ya se tiene 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 uti-

*Corresponding author

lidad, 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.

2. Aproximaciones

Lo que queremos hacer es utilizar sustancias y sistema periódico (PS) como entrada para un algoritmo. Digamos, por ejemplo, que la sustancia R-Na existe. Luego queremos preguntarle al algoritmo: Dado eso, cual es la probabilidad de que R-Fe exista? (Fe o cualquier elemento).

Nuestro algoritmo se encarga entonces de calcular la siguiente probabilidad:

$$P(existeR - X|existeR - Y + PS) \tag{1}$$

Con X cualquier elemento conocido en la época, Y un elemento tal que R-Y existía en la época y PS es el sistema periódico de esa época.

Lo que estamos buscando es un algoritmo 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 ser alta, si uno de los compuestos existe. Esta información de similaridad debería poder encontrarse, por supuesto, en el PS.

2.1. Input

El input puede ser un esquema de la tabla periódica, 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_i existe para todo i, en la época considerada. Entonces en nuestra representación, en la posición de cada uno de los elementos de A en el PS 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 del PS (PSR). En caso que R-Y exista en esta época, el label (y) es igual a 1, de otra manera es igual a 0.

Aún hay que pensar en los valores asignados a las casillas del PSR, pues hay 4 casos:

- Si R-X existe
- Si R-X no existe
- Si R-Y es el compuesto problema (al que calculamos la probabilidad)
- Si la casilla no corresponde a ningún elemento

Esto puede evaluarse a medida que se empiece a desarrollar código y se hagan pruebas, para ver qué funciona mejor.

2.2. Generación de los datos

Para cada fragmento R diferente en el conjunto de sustancias, se construyen PSR y se llenan con 0s. Luego llenamos con 1s para cada elemento X tal que RX exista. Esto da como resultado N(R) tablas. N(R) es la cantidad de fragmentos R diferentes en el conjunto de sustancias considerado.

Después de esto se pueden hacer 2 cosas:

- Iterar por todos los elementos y preguntar si R-A_i existe o no.
- Seleccionar una muestra aleatoria de elementos para realizar la misma pregunta. En este caso creo que deberían estar asegurados todos los compuestos que sí existen. Esto es, si el compuesto R-B existe, entonces B debe hacer parte con certeza de esta muestra aleatoria. Esto último puede discutirse pues también es necesario tener una cantidad de datos de prueba (test set).

Cualquiera de estas aproximaciones que se tomen, ambas llevan a escribir un -1 en la posición del elemento en el PSR, y un 0 ó 1 como label dependiendo si el compuesto existe o no.

Esto claramente aumenta la cantidad de datos disponibles. En principio puede llegarse a un dataset con N(R) * $|A_i|$ elementos, donde A_i es el conjunto de los elementos conocidos en la época.

2.3. Variantes

Se pueden considerar otras variantes a esta aproximación, que en mayor o menor medida pueden contribuir a la expansión del dataset. Estas variantes vienen de la siguiente consideración: pueden existir compuestos en los cuales un elemento esté más de una vez, esto es, compuestos de la forma RX_n con $n \geq 1$. En esta sección vamos a considerar la generación de los datos como ya se explicó, teniendo en cuenta esta consideración.

2.3.1. X $\sim$ Y si n == m

En la aproximación más simple, 2 compuestos se encuentran en la misma tabla sí y sólo sí existen compuestos R-X_n y R-Y_m, con n == m. La cantidad máxima de tablas que se obtiene de esta manera es $N * \langle M \rangle$; donde N es el

número de compuestos en el dataset y $\langle M \rangle$ es el número promedio de elementos por compuesto. Por supuesto, la cantidad de tablas diferentes es menor a esto, pues para que la condición de arriba se cumpla deben existir al menos dos compuestos que compartan R y sólo difieran por un elemento.

2.3.2. Considerar distribuciones del subíndice

Esta aproximación es probablemente la más complicada en términos de implementación, sin embargo tiene el potencial de expandir en gran manera la cantidad de datos disponibles y mejorar la calidad de las comparaciones. La consideración es la siguiente:

Suponga que tiene un compuesto R-X_n, con $n \geq 1$. Bajo esta aproximación, podremos considerar como un nuevo R′ la combinación RX, de manera que tenemos un "nuevo compuesto" RX-X_{n-1} = R′-X_{n-1}, y seguir la aproximación convencional para reemplazar X y evaluar similitud. 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. Bajo este método se obtienen un máximo de $N * \langle M \rangle * \langle n \rangle$, donde N es la cantidad de compuestos, $\langle M \rangle$ la cantidad promedio de elementos por compuesto, y $\langle n \rangle$ es la cantidad promedio de átomos de un elemento por compuesto.

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. Desde el punto de vista químico, por supuesto, esta aproximación explota de mejor manera algunas similaridades, en principio obvias, como la de los halógenos, o los alcalinos, etc. Sin embargo tiene el potencial de encontrar nuevas -menos obvias- similaridades.

Por ejemplo, las fórmulas $C_6BrCl_2H_3$ y $C_6Cl_3H_3$ no tienen nada en común bajo la primera aproximación, pues la primera tiene 4 elementos y la segunda 3, por lo que son incomparables. Bajo el método que discutimos ahora, ambas fórmulas pueden compararse, y de hecho dan lugar a conclusiones importantes. Esto es, los compuestos 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 la clara ventaja de este método por encima del primero.

2.4. Resultados de 2.3

Los dos métodos descritos anteriormente fueron implementados. El código hace uso de 2 funciones auxiliares, en una de estas se extraen los elementos únicos existentes en el conjunto de datos, y en la otra se convierten los compuestos del conjunto de datos en vectores para el tratamiento posterior como ya se describió.

La figura 1 ilustra las representaciones objetivo que se describieron anteriormente. Esta fue generada utilizando el código mencionado con un conjunto de datos preparado artificialmente.

Figura 1: Representación de la tabla periódica generada con los métodos de la sección 2.3. (Arriba/abajo) representación incluyendo/sin serie de lantánidos.

En la figura 1, ambas representaciones indican 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? Como ya se mencionó, esto debe ir acompañado de una etiqueta que marca la respuesta real a esta pregunta, nuevamente basado en el conjunto de datos. Esto convierte la tarea de leer la tabla periódica en una tarea de aprendizaje supervisado.

Al analizar los resultados de esta prueba, se encuentran similaridades entre, por ejemplo, el H y K debido a su presencia en los compuestos H_2O y KOH (bajo el segundo método), además de la ya mencionada en el sistema con Br y Cl, entre otros. Esto sugiere que es el método más apropiado. En general se encuentran similitudes con sentido (desde lo que se enseña comunmente sobre la TP), además de esto se espera que las similitudes "raras"sean

suprimidas estadísticamente, esto es, si la similitud es demasiado extraña entonces ocurre muy pocas veces y otras similitudes más comunes sobresalen. Esto puede ser una ventaja o una desventaja según el contexto, sin embargo esto podría evaluarse después de obtener los primeros modelos.

2.5. Ideas para sistema predictor

La idea inicialmente es utilizar redes neuronales para aproximar la función descrita por la ecuación 1. La arquitectura a usar es uno de los grandes problemas debido a que no existe una única manera de obtener una óptima y usualmente se hace uso del ensayo y error. 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, estas imágenes deben tener un tamaño fijo, lo cual no es un problema en nuestro caso pues la tabla periódica tiene un tamaño (en pixeles) igual para todas las imágenes.

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. Esto es claro en el siguiente 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 no puede pretenderse obtener propiedades de los compuestos únicamente a partir de la representación hasta aquí formulada.

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. Esto es similar a lo que se muestra en [2], donde se presenta un concepto similar (figura 2). En nuestro caso, sin embargo, uno de los "caminos" paralelos sería una CNN que recibe la representación de tabla periódica aquí propuesta, mientras que el otro sería una DNN que recibe información sobre el fragmento R correspondiente.

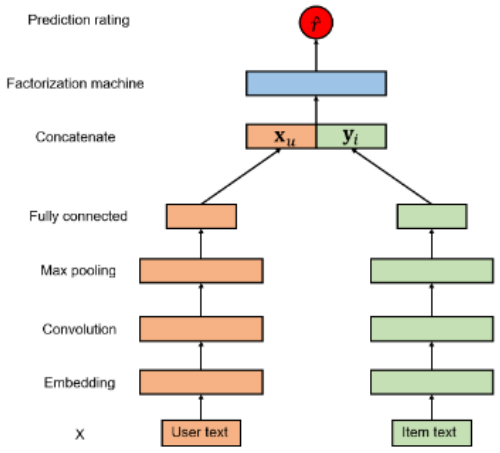

Figura 2: Arquitectura propuesta de red neuronal. Imagen obtenida de [2].

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.