Campos Aleatorios Condicionales: Optimización de parámetros del modelo para la clasificación de genes

Fernanda G. Alejandro L. Brenda S. Alejandra S. Elizabeth S.
Clase de Bioinformática I
Módulo Minería de textos

03 de octubre de 2019

1. Resumen

Los campos aleatorios condicionales (CRFs) nos permiten clasificar objetos específicos. En este caso fue usado para la clasificación de genes en resúmenes de artículos científicos. Buscamos analizar los parámetros que ayudan a subir el desempeño de un modelo. Para esto, ejecutamos diferentes versiones de dicho modelo y definimos ciertos parámetros que nos permitieron obtener una mejor puntuación. Además, se demostró que la proporción de datos otorgados para el entrenamiento tiene un gran peso en los resultados finales. Por último, encontramos parámetros que nos ayudaron a subir la exhaustividad, es decir, disminuyó la tasa de falsos negativos; sin embargo, disminuyó la puntuación de precisión, es decir, aumentó la tasa de falsos positivos, por lo que decidimos buscar el punto en el cual el funcionamiento del modelo se maximice, al ser minimizadas tanto la tasa de falsos positivos, como la de falsos negativos.

2. Palabras Clave

Aprendizaje máquina, campos aleatorios condicionales, datos no estructurados, minería de textos, exhaustividad, flat score, precisión.

3. Introducción

El campo del aprendizaje máquina puede definirse como la disciplina científica que crea sistemas que "aprenden" automáticamente y que pueden mejorar con la experiencia para ayudar a los seres humanos en el análisis de grandes y complejos conjuntos de datos.

El aprendizaje máquina se divide en dos campos. El primero es aprendizaje máquina supervisado, que utiliza datos clasificados y etiquetados correctamente para su análisis. Por el contrario, el aprendizaje máquina no supervisado se alimenta de datos no etiquetados a partir de los cuales aprende.

Los métodos de aprendizaje máquina supervisado enfocados a la clasificación de datos consisten en tres etapas principales; la creación de un algoritmo que, para el desarrollador pueda cumplir las necesidades del análisis que desee realizar; el otorgamiento al algoritmo de una serie de datos de la cual se conozca su clasificación, para que pueda aprender; y la evaluación de la máquina con una serie de datos con clasificación desconocida para la predicción de las clases. De ser correcta la predicción, podemos asumir que la máquina funciona y así probarla con datos completamente nuevos.

A continuación trabajaremos con un CRF, un método de aprendizaje máquina supervisado, con el cual compararemos los cambios sobre los resultados al modificar los valores de distintos parámetros. Intentaremos encontrar los parámetros óptimos en los cuales, el desempeño del modelo sea el mejor.

4. Métodos

Los campos aleatorios condicionales son modelos matemáticos estocásticos que funcionan para la segmentación y el etiquetado de secuencias, en este caso, de palabras de archivos de textos. El modelo matemático se basa en una variable aleatoria X, que corresponde a secuencias a etiquetar, y en una variable aleatoria Y, que corresponde a la etiqueta para cada X. Por lo tanto, la predicción de la etiqueta correspondiente a cada valor de X depende de éste mismo. De este modo, se puede describir un modelo condicional:

$$p = (Y|X) \tag{1}$$

Para llevar a cabo el entrenamiento del modelo, se nos proporcionaron los resúmenes de 455 artículos científicos

Tabla 1: Parámetros modificados en cada versión				
Parámetros	**Default (v3.0)**	**V3.1**	**V3.2**	**V3.3**
Número de palabras consideradas antes y después de la palabra en cuestión	Una antes y una después.	Dos antes y dos después.	Dos antes y dos después.	Dos antes y dos después.
Considerar las últimas letras en minúscula	No	Sí	Sí	Sí
C1	0.5	0.1	0.03	0.5
C2	0.05	0.1	0.04	0.05

Tabla 1: Parámetros que fueron modificados en las versiones realizadas. El default corresponde a la versión antes de que se realizaran las modificaciones. Se modificaron 4 parámetros: el número de palabras consideradas antes y después de la palabra en cuestión, la consideración de la ultima letra minúscula, C1 y C2.

sobre fibrosis pulmonar idiopática previamente *tokenizados* y lematizados. Los artículos se dividieron en datos de entrenamiento y datos de evaluación, con proporciones de 70 % y 30 %, respectivamente. De este modo, el modelo fue entrenado con artículos cuyas palabras ya se encontraban etiquetadas como genes y no genes. Cuando se le evaluó con el conjunto de datos del 30 %, se compararon las clases predichas con las clases verdaderas, para entonces conocer los resultados finales del modelo.

El modelo fue evaluado con tres métricas distintas; precisión, que indica cuántos de los objetos predichos en una clase son verdaderamente de esa clase; exhaustividad, también llamado *recall*, que indica cuántos objetos de una clase fueron predichos dentro de esa clase, y F1 score, que pondera ambas medidas mencionadas anteriormente por una media armónica. Para la evaluación del modelo, fueron consideradas las tres métricas mencionadas para ambas clases posibles, es decir, genes y no genes, así como una medida que evalúa al modelo en su totalidad, F1 score flat.

Para el entrenamiento del modelo, se nos proporcionaron tres algoritmos distintos. Al ejecutarlos sin alteración del código, los valores de F1 flat score fueron de 0.8414, 0.842 y 0.8579 para los algoritmos 1, 2 y 3, respectivamente. Por lo tanto, decidimos partir del algoritmo número 3, al cual le realizamos diversas modificaciones con la finalidad de mejorar sus puntuaciones con respecto al algoritmo y a las puntuaciones de referencia, es decir, el algoritmo 2, con puntuación de 0.842.

Una vez habiendo decidido el algoritmo a utilizar, realizamos modificaciones en éste para definir cuáles fueron los parámetros que tenían un mayor impacto en la evaluación final del modelo. Observamos que los parámetros más significativos fueron la consideración de letras minúsculas al final de las oraciones, el número de palabras tomadas en cuenta antes y después de la palabra en cuestión y la modificación de los hiperparámetros c1 y c2. La consideración de las letras minúsculas se explica debido a que hay una alta probabilidad de que si la última letra de una palabra es una mayúscula, un número, o no es una letra del alfabeto anglosajón, sea un gen. El número de palabras a juzgar antes y después con respecto a la palabra en cuestión considera que dos palabras otorgan una mayor información que únicamente una. Finalmente, c1 y c2 corresponden a las regularizaciones de los parámetros $\lambda 1$ y $\lambda 2$, respectivamente, es decir, los pesos asignados a las funciones del algoritmo, a partir de los cuales éste va a aprender. Estos parámetros se estiman mediante métodos de máxima similitud *Tabla 1*.

Con la decisión del algoritmo a analizar y los parámetros significativos a cambiar, iniciamos con la modificación del código. Los objetivos fueron lograr obtener valores altos en la evaluación del modelo, con énfasis en la evaluación con respecto a los genes. Todos los códigos evaluados y los datos utilizados se encuentran en el siguiente repositorio: https://github.com/fgarcia27/MineriaBioinfo

5. Resultados

Como se puede observar en la tabla 1, respecto al tercer algoritmo, creamos tres nuevas versiones con distintas modificaciones de los tres parámetros que se observó eran los que tenían un mayor impacto en el desempeño del modelo. En la tabla 1 se pueden observar los valores que se usaron para cada parámetro en las tres distintas versiones.

Para cada versión nueva creada, ejecutamos el códi-

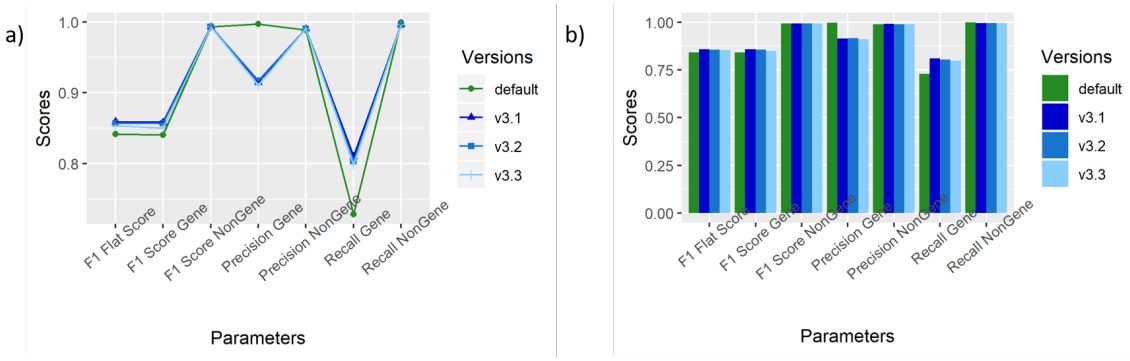

Figura 1: Gráfica de las puntuaciones de los 7 parámetros de evaluación por cada versión del modelo. (a) Diagrama de dispersión. (b) Gráficas de barras.

go cinco veces, hicimos anotaciones de los valores de las siete medidas de evaluación y obtuvimos las medias. Esto con la finalidad de evitar fluctuaciones fuertes en los valores de las medidas de evaluación debido a factores e inicializaciones de variables al azar. Posteriormente, procedimos a comparar las tres versiones nuevas entre ellas y con respecto al algoritmo estándar *(Figura 1)*

Como se puede observar en las gráficas mostradas en la figura 1, los tres modelos modificados tuvieron un mejor desempeño total con respecto al código original, lo cual es observable con el parámetro F1 flat score. Asimismo, para determinar el modelo que tuvo el mejor desempeño de todos, nos enfocamos en la clasificación de los genes, por lo que aquél modelo con los valores más altos de genes se definió como el mejor. Definimos que el modelo mejor entrenado fue la versión 1, pues tanto para precisión como para exhaustividad de genes, sus valores fueron los más altos. Por ende, su F1 score para genes también lo fue.

Una vez elegido el mejor modelo, evaluamos el impacto del tamaño de los datos de entrenamiento en el desempeño del modelo. Evaluamos el modelo original y la versión 1 con una fracción del 35 % para los datos de entrenamiento *(Figura 2)*. Al disminuir los datos de entrenamiento, podemos observar que el desempeño de ambos modelos disminuye. Los valores de todos los parámetros de las versiones con el 35 % de datos para entrenamiento disminuyeron respecto a su versión correspondiente entrenada con el 70 % de los datos. Debido a ello, podemos observar que a pesar de que la versión 1 es mejor que la versión original, la versión original entrenada con el 70 % de datos tiene un mejor desempeño total que la versión 1 entrenada con el 35 % de los datos, lo cual nos muestra una fuerte relación entre el desempeño del modelo y la cantidad de datos con la que fue entrenado.

6. Discusión

Creemos que la primera versión tiene el mayor F1 score debido a que toma dos palabras antes de la considerada, y dos después, esto otorga información más relevante acerca de la clase de la misma. Asimismo, la consideración de caracteres en minúsculas al final de la palabra es buena para la definición de muchas palabras que probablemente no sean genes, y modificaciones realizadas sobre c1 y c2 modifican la rapidez de convergencia de los valores de λ, por lo que la facilidad de aprendizaje del modelo es mayor.

Debido a esto, y a que la consideración de minúsculas en la última palabra fue una modificación realizada en todas las versiones, la exhaustividad en las tres aumentó considerablemente con respecto al modelo original. Por esto, se están recuperando más genes de los que se recuperan sin considerar que una palabra con un número, una mayúscula o un carácter no anglosajón es muy probablemente un gen.

Asimismo, creemos que la razón por la que los valores de precisión del gen disminuyeron tanto con respecto al modelo original, se debe a que la consideración de las letras minúsculas al final de las palabras ocasionó que el modelo generalizara fuertemente situaciones en las que esta condición no se cumple, por lo que clasificó como genes muchas palabras que no lo eran, por tener características similares a éstos. Del mismo modo, creemos que los valores modificados de c1 y c2 ocasionan una mejora en el modelo, debido a que se modifica la inicialización aleatoria de los valores de , y es muy probable que los valores otorgados le faciliten la convergencia de éstos al modelo mediante descenso de gradiente.

Finalmente, en la figura 2, somos capaces de observar que el número de datos de entrenamiento influye fuertemente en el desempeño final del modelo. Las versiones entrenadas con el 35 % de los datos tienen una menor habilidad de clasificación con respecto a las versiones entrenadas con el 70 % de los datos. Esto se debe a que, en conjunto, el número de características y situaciones es-

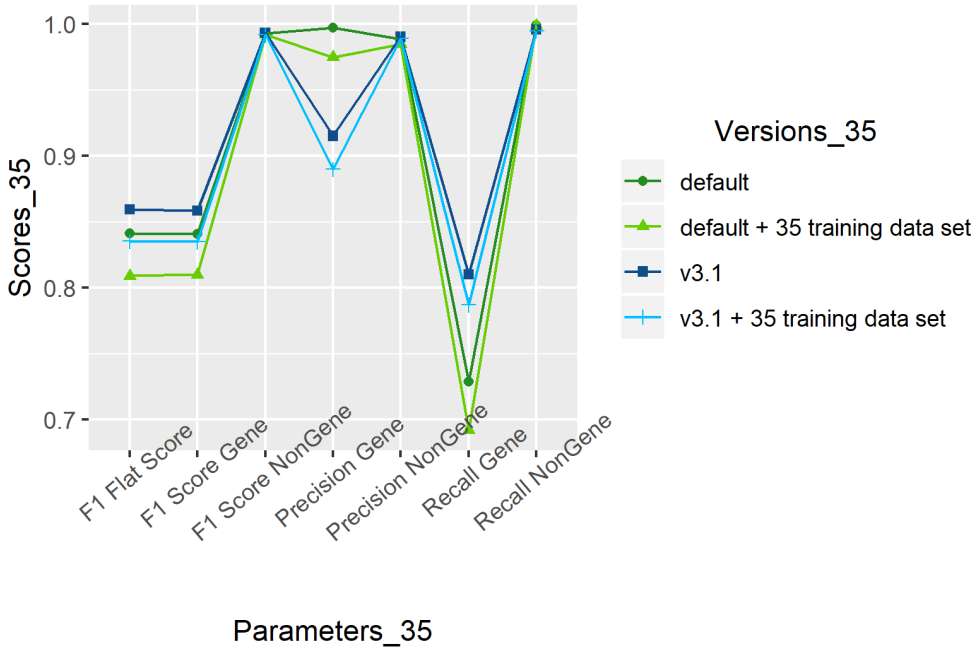

Figura 2: Diagrama de dispersión de la versión original y la versión 3.1, ambos entrenados con conjuntos de datos del 70 % y del 35 %.

pecíficas "aprendidas" por el modelo, es mucho menor y tiene una menor afinación cuando el conjunto de datos es pequeño.

A pesar de que un conjunto pequeño de datos de entrenamiento disminuye el desempeño del modelo, no es recomendable que el conjunto de datos de entrenamiento supere el 80 %, pues es probable que ocurra un proceso denominado sobreajuste, esto es, que el modelo se ajusta a características muy específicas, por lo que tiene un desempeño muy alto en datos similares, pero en datos distintos, disminuye considerablemente. No fuimos capaces de identificar si esto estaba ocurriendo en nuestro modelo, o si ocurría al aumentar la proporción de datos de entrenamiento, debido a que no contábamos con datos nuevos para comprobarlo. Por lo tanto, para esta finalidad, se requeriría una evaluación más exhaustiva.

7. Conclusión

El aprendizaje máquina tiene aplicaciones diversas, dentro de las cuales resalta la minería de textos. Actualmente, existe una gran cantidad de información en forma no estructurada (sin un formato ordenado), por lo que la obtención y procesamiento de ella es tedioso y poco práctico. Estructuras como las bases de datos brindan este ordenamiento, pero para ello es necesaria la obtención de información.

Los CRFs son una herramienta útil para la clasificación y por tanto la extracción de información para ser estructurada posteriormente. Esta es una de las muchas utilidades prácticas de la clasificación y por lo tanto de los CRFs.

En nuestra investigación, pudimos demostrar que hay parámetros que mejoran la eficiencia de los CRFs, arrojando mejores resultados de clasificación. Lo anterior nos sirve para ubicar mejor los posibles cambios necesarios para facilitar la futura tarea de estructuración de datos.

8. Referencias

- Libbrecht, M. W., Noble, W. S. (2015). Machine learning applications in genetics and genomics. Nature Reviews Genetics, 16(6), 321.
- allach. (2004). Conditional Random Fields: An Introduction.
- F. Sha and F. Pereira. Shallow parsing with conditional random fields Proceedings of Human Language Technology, NAACL 2003, 2003.
- He, Y., Kayaalp, M. (2008). Biological entity recognition with conditional random fields. AMIA ... Annual Symposium proceedings. AMIA Symposium, 2008, 293–297.