El desarrollo de un modelo de *machine learning*, terminó por derivar en el estudio y aplicación de múltiples áreas de las matemáticas, profundizar en temas como la simulación, modelos estructurados y probabilidad bayesiana, fue una tarea altamente edificante. Fue muy interesante el reto que representó el entendimiento de un modelo tan complejo como el presentado en este trabajo y fue aún más gratificante que el modelo funcionara tan bien como lo hizo.

0.1. Consideraciones finales del modelo

Este trabajo, como todo modelo estadístico, no está exento de contratiempos, limitaciones y consideraciones que se deben tomar en cuenta a la hora de aplicarlo. Aunque, en general, es un buen modelo de clasificación supervisada, siempre hay que tomarse los resultados con cautela crítica y ponerlos a prueba. Los modelos estadísticos, tanto por sus características como por el uso de datos muestrales, son aproximaciones a la realidad y deben ser usados con criterio. Sin embargo, es innegable que se estén convertido en herramientas, útiles y necesarias, en la mayoría de los ámbitos de la civilización contemporánea, como lo son las finanzas, la ciencia y la salud.

Convergencia y sus implicaciones

En particular, este capítulo busca revisar las limitaciones y contratiempos que podrían surgir. Como primer punto, repetido a lo largo del trabajo, se revisa la convergencia del modelo pues esta es fundamental. Lograr cadenas siempre convergentes, estables y que tengan la distribución posterior deseada es difícil. Esto, pues los métodos de simulación bayesianos, dependen de parámetros, variables, algoritmos y generadores de números aleatorios, y sería inútil pedir que todos los modelos convergencia a la perfección. Los algoritmos MCMC aunque complejos y hasta cierto punto misteriosos, se deben entender y *afinar* para la aplicación en concreto.[1] Sin embargo, no por la dificultad, se debe obviar la convergencia, de otra forma, se estaría tratando de acertar (dejando todo a la suerte) a los valores subjetivos y sesgados del estadista. Se debe de tener cierto criterio para aceptar desviaciones y mejor aún, entenderlas y tratar de corregirlas. En general todos los ejemplos presentados en este trabajo convergieron de forma relativamente buena, pero sobre todo *replicables*, lo cual indica que si existe un patrón que el modelo está encontrando y no fue un golpe de suerte estadístico el encontrar las regiones de separación.

La convergencia del modelo, ahora, vista desde el punto de vista computacional y no tanto bayesiano, es uno problema más importantes de los que sufre aún el algoritmo. No es raro, que al aumentar d, algunos parámetros empiecen a tener problemas de

1. Al principio de este trabajo, se consideró usar un algoritmo de cadenas de Markov *Hamiltonianas*, que combina ideas de física para lograr estimaciones más robustas y con menor correlación entre los parámetros. Sin embargo, dada la complejidad en su aplicación al modelo, se optó por usar algoritmos más sencillos y fáciles de implementar directos del trabajo de **albert1993bayesian**

escala y terminen por divergir. El ejemplo 6, sobre el que se realizó todo el análisis de convergencia, sufre justamente de este problema. Si la cadena fuera más grande, el parámetro $\hat{w}_2$ hubiera seguido creciendo y el algoritmo (que depende de inversión de matrices) hubiera terminado por caer en errores de condicionamiento numéricos. Sin embargo, la fuente del error es bien conocida; si se revisa la ecuación de los residuales parciales **??** aplicados al modelo, se ve claramente cuando si $\beta_{j^*} \to 0$, los residuales parciales de esta variable o dimensión $r_{j^*} \to \infty$ y por lo tanto el vector w_{j^*}. Esta falta de ortogonalidad entre parámetros β y $\mathbf{w}$ es complicada de corregir, usualmente, se deja de usar β enteramente y se trata de capturar toda la información a través de $\mathbf{w}$ como en los GAM tradicionales. Sin embargo, ningún algoritmo es mejor que otro y los resultados que se lograron fueron suficientemente aceptables.

Calibración de los parámetros y velocidad del algoritmo

Aunque se podría pensar que al mover M, J y K a discreción del estadista se están sesgando los resultados, en realidad es solo una consecuencia de haber escogido un modelo tan complejo y estructurado. En prácticamente ningún modelo estadístico, incluso en los no paramétricos, se puede dejar todo al algoritmo y que esta encuentre el modelo perfecto. Siempre habrá un parámetro o variable que se debe de *afinar*, lo cual introduce una dimensión subjetiva al modelo. La diferencia para este trabajo, es que se tienen que afinar algunos parámetros más. Sin embargo, este proceso de calibración, se puede hacer de tal forma que no sea por *fuerza bruta*, solamente buscando obtener resultados; por el contrario, la calibración debe ser un proceso

analítico, que analiza el porqué esa selección particular de parámetros no funcionó y modificarlos en respuesta. En la practica sin embargo, y con excepciones contadas,[2] la selección de $M,\ J$ y K para las bases de datos sencillas era prácticamente trivial y el modelo siempre capturaba el patrón, con diferentes tipos de fronteras; como se vio en los primeros ejemplos del análisis de sensibilidad de la sección **??**.

Es curioso notar, que aunque el modelo sea complejo y pueda crecer rápidamente en el número de parámetros modificando $M,\ J$ y K, la velocidad del algoritmo es bastante buena. Gracias a las optimizaciones realizadas en los cálculos parciales y el uso de distribuciones conjugadas, la simulación de un gran número de parámetros es relativamente trivial. Fuera de esos casos, prácticamente todos los modelos corridos, se terminaron en un minuto o menos. Aquello que hace que el algoritmo sea más lento, usualmente es aumentar d o escalar n varias ordenes de magnitud. Gracias a la fácil disponibilidad y aplicación del paquete *bpwpm*, se exhorta al lector probarlo sobre diferentes datos y problemas, ya que sería interesante verlo aplicado en otros contextos y datos. Además, el algoritmo se puede ir mejorando con contribuciones externas.

Otro factor importante que influye en la velocidad del algoritmo es el uso de un paradigma bayesiano en el entrenamiento. Esta decisión se toma más que nada por cuestiones personales, ya que la filosofía bayesiana de *actualización del conocimiento* resuena mucho con aquella del autor. Sin embargo, el paradigma frecuentista es muy valioso por si mismo y en este caso (usando métodos tradicionales de estimación)

2. Usualmente para casos límite cuando $K = 0$

hubiera logrado que el algoritmo, fuera casi instantáneo par aun número enorme de parámetros, sin embargo, este enfoque hubiera requerido hacer un trabajo completamente diferente, con sus pros y sus contras.

0.2. Posibles mejoras y actualizaciones

La fuerza del modelo recae en el gran número de componentes que tiene, sin embargo, este número también le otorga cierta *flexibilidad*, no tanto en la estimación, sino en su estructura. Cada parte que contiene, se puede modificar de una infinidad de formas, haciendo el modelo más complejo o más sencillo, más robusto o para otras aplicaciones. Al final, este no es infalible y siempre hay espacio para mejorar.

La primer y más urgente mejora que se propone explorar, es la de incorporación de un método para la *selección de variables*. El enfoque de la estadística frecuentista, especialmente para modelos de regresión, es buscar aquellas variables *más significativas* para la predicción de la respuesta. Existen procedimientos iterativos *hacia adelante y hacia atrás*, que exploran el espacio de 2^d modelos posibles y encuentran el mejor usando criterios análogos al de la función log-loss usada en este trabajo.[3] Los métodos de ML más recientes son especialmente efectivos en este ámbito; sus algoritmos recaen en usar cantidades enormes de información con múltiples variables ($d >> 0$) para hacer predicciones robustas al entrenar miles de parámetros. Bajo un paradigma bayesiano la selección de variables también se puede tratar bajo

3. Usualmente el criterio de Akaike

esta óptica. Los métodos más usados, incorporan otra serie nueva serie de variables auxiliares (usualmente indicadoras), cuya función es *detectar* cuando una variable es relevante o no. A estas variables, también se les da un tratamiento bayesiando y son estimadas por los mismos algoritmos MCMC a la par de todas las demás (**o2009review**).

Para este trabajo sin embargo, esta selección de variables se hizo de manera manual (y subjetiva hasta cierto punto) tomando únicamente aquellas que se consideraban importantes o útiles, derivado de una exploración a priori de los datos. La urgencia de incorporar esto al modelo, se debe a que la selección de variables, no solo se realiza en afán de simplificar los modelos, sino por una razón computacional de convergencia. Por lo mismo que se discutió arriba, cuando una β_j era cercana a cero, las cadenas tendían a diverger, haciendo la estimación imposible. Asimismo, la colinealidad entre variables puede exacerbar este problema, volviendo la identificación de variables relevantes una cuestión todavía más urgente para el modelo. Por lo pronto, para d entre 1 y 4, el modelo funciona bien, solamente se debe tener en mente la longitud de las cadenas.

La siguiente modificación interesante está en la selección automática de posiciones nodales. La principal razón por la que no se logró estimar perfectamente el ejemplo del *yin-yang* se debe a que los nodos se concentraban hacia el centro donde hay más datos y no en los pequeños círculos donde se necesitaban. Esto viene derivado de que hasta el momento, sus posiciones se eligen en los cuantiles de los datos. Como se mencionó, el mismo trabajo rector de este trabajo **mallik1998automatic**,

considera un método para realizar esto, pero implicaría usar métodos más avanzados en el algoritmo MCMC pues las dimensiones cambian de forma constante. Balancear esa capa adicional con la estimación de todos los parámetros, latentes y no latentes, salía del enfoque de este trabajo y hubiera mejorado marginalmente las estimaciones presentadas. Sin embargo, vale la pena tomarlo en cuenta para el futuro.

Otra modificación considerada es volver el algoritmo de muestreo Gibbs en algo menos rígido. Como se menciona en el Capítulo **??**, se toman distribuciones conjugadas para el proceso de aprendizaje bayesiano pues simplifica mucho la derivación de la ecuación (**??**), conviriendola en (**??**) lo cual permite que el muestreo sea sencillo, requiriendo únicamente álgebra lineal y simulaciones de distribuciones normales multivariadas. Aunque el supuesto no es malo, sería bueno poder incorporar distribuciones a priori arbitrarias, para poder reflejar conocimiento previo de la base de datos o información de expertos. Hacer esta modificación sin embargo, si requeriría de cambiar sustancialmente todo el algoritmo, y por ende las derivaciones, Asimismo, se estaría obligando a usar paquetes de software que permitan estimaciones más generales como las librerías `STAN` o `BUGGS` que, aunque son excelente herramientas bayesianas, no eran el lenguaje que se planeaba usar para este trabajo.

Se hace notar que el algoritmo se implementó en el software estadístico `R`. Aunque `R` tiene múltiples ventajas en el uso de estructuras y cálculo de medidas estadisticos, no es el lenguaje más veloz pues corre a un nivel muy alto. Si se pensara usar el algoritmo para aplicaciones más robustas, se recomendaría usar lenguajes de nivel más bajo como `C++`.

Como última modificación, se considera que si se usara una expansión de bases diferente, sería posible mejorar tanto la velocidad, como la precisión del algoritmo más allá de los nodos. La expansión en bases truncadas es buena y en la práctica funciono muy bien, sin embargo, es computacionalmente lenta. Si se incorporara el cambio en la posición de los nodos sería forzoso recalcular las matrices Φ_j múltiples veces, haciendo que el algoritmo se volviera lento. Haciendo un cambio de bases, se puede usar un conjunto de b-splines que representen exactamente el mismo polinomio sustancialmente más rápido. Asimismo, esta modificación permitiría incorporar los *splines naturales* que no fluctúan tan rápido, más allá de la frontera.

Estas capacidades adicionales, robustecerían en gran forma al modelo y lo harían una herramienta muy poderosa. Si se pensara en usar el modelo para aplicaciones a gran escala, con miles de datos más, sería vital incorporarlas. Sin embargo, para efectos de este trabajo, muchas de estos problemas, se pueden superar de formas sencillas y no fueron en realidad contratiempos para los ejemplos presentados.

0.3. El aprendizaje de una máquina

El mundo de la estadística computacional ha sido revolucionado en las últimas décadas. Gracias a los grandes estadistas como los citados, que han visto más allá de los métodos tradicionales, es que se han dado avances astronómicos en las posibilidades. Eso, aunado al aumento exponencial en las capacidades de cómputo, los modelos, se han vuelto cada vez más poderosos y útiles en la vida real.

Algunos de los métodos de ML no son más que modelos GLM como el presentado, que se corre miles de veces sobre bases de datos gigantezcas, donde existen capas de regresiones y un sinfín de parámetros por estimar. Las redes neuronales por ejemplo, son regresiones sucesivas entre *neuronas* de información, que no son otra cosa más que variables latentes z intermedias. Cada capa de neuronas, va captando patrones subyacentes de los datos. Las neuronas, se dice que se activaron cuando la función liga, después de colapsar dimensiones, rebasa cierto umbral. Este proceso se corre miles de veces entre miles de neuronas[4] logrando detectar patrones cada vez más complejos. Si para este trabajo se usan muchos índices, en los textos de ML se usan incluso más. Al final, fuera de las capacidades de estos modelos y su complejidad, la gran mayoría, son *regresiones glorificadas* que se basan en los mismos principios que el presentado en este trabajo. Por lo mismo, valía hacer una exploración a fondo de uno modelo análogo.

La fuerza que han adquirido las técnicas de ML en los últimos años, es que han logrado romper con muchos de los paradigmas tradicionales. Estos responden preguntas como: ¿se pueden aplicar modelos estadísticos a imágenes y sonidos? ¿por qué restringirse a dos categorías en la respuesta? y ¿a cuantos datos y variables se puede aplicar?. En general, las respuestas son más que positivas, tanto, que dispositivos de de uso diario, usan estos modelos para clasificar fotos, recopilar información o entender el lenguaje hablado. Los modelos han sido clave para el desarrollo de un mundo cada vez más futurista y probablemente seguirán avanzando en sus capacidades. Entenderlos y poder analizarlos, se vuelve clave pues, al final, se le está dando

4. Usualmente de manera frecuentista.

un nuevo sentido a lo que implica *que una máquina aprenda*.

Con este trabajo, además de desarrollar el modelo, se buscó dar una base teórica y técnica de las posibles extensiones del *aprendizaje de máquina*. El autor, espera que se le haya dado un mejor contexto a la también llamada *Inteligencia Artificial*, lo cual, se espera se haya visto que no es más que estadística computacional llevada al límite.