El desarrollo de un modelo de aprendizaje de máquina, derivó, en el estudio y aplicación de múltiples áreas de la estadística y las matemáticas. Siguiendo una vez más el precepto de **box1976science**, el modelo se flexibiliza en respuesta a una necesidad práctica y no por si mismo. Sin embargo, es interesante el reto que esto representa, así como el entendimiento profundo de modelo como el presentado en este trabajo o muchos otros usados en la práctica actual, asimismo, es altamente gratificante observar los buenos resultados. No obstante, el modelo se puede mejorar significativamente, además de que restan algunos temas en los que vale la pena profundizar. Este capítulo busca resumir las posibles limitaciones y contratiempos que podrían surgir en la aplicación del modelo que el autor anticipa.

0.1. Consideraciones finales sobre el modelo

Ventajas con respecto a procedimientos tradicionales

Como se hizo notar a lo largo del trabajo, el *bpwpm* no es más que la combinación de dos trabajos claves, la flexibilización de la función de predicción η por parte de **mallik1998automatic** y la definición de un modelo probit (que induce un algoritmo de muestreo Gibbs) por parte de **albert1993bayesian**. Sin embargo al combinar sus trabajos y extender sobre sus posibilidades, el modelo logra diversas fronteras de predicción, imposibles para modelos con fronteras más rígidas como se ilustró en la sección **??**. Asimismo, el que el modelo sea fácil de implementar, calibrar y validar lo hace pedagógico, logrando ilustrar conceptos que en apredizaje estadístico pudieran

parecer confusos. Otro beneficio, es la utilización de un paradigma bayesiano de apredizaje. Este paradigma, permite poner modelos en producción en aplicaciones reales donde se reciben constantemente nuevos datos. Si se cree que los parámetros no son estáticos en el tiempo, conforme se reciben nuevas observaciones se pueden hacer predicciones en tiempo real y al mismo tiempo refinar los parámetros en presencia de nueva la evidencia.

Convergencia y sus implicaciones

Lograr cadenas siempre convergentes, estables y que tengan la distribución posterior buscada es complejo por dos razones. Primeramente pues, aunque el muestreador de Gibbs garantice la ergodicidad, si se tienen regiones de baja probabilidad, el algoritmo podría tomar un tiempo casi infinito en alcanzarlas, **robert2004monte**. Segundo por el componente numérico del algoritmo, esto pues la mayoría de los métodos bayesianos de simulación dependen de generadores de números aleatorios, además, se podrían dar errores de estimación por problemas de desbordamiento binario. Por ejemplo, no es raro que el algoritmo caiga en errores de precisión de máquina al tener términos de orden mayor ($M >> 0$) que crecen rápidamente fuera de $\mathcal{X}^d$. Sin embargo, todos los modelos presentados en este trabajo son replicables al fijar la semilla del algoritmo generador, e incluso, si este se cambia, los parámetros siguen convergiendo a los valores puntuales presentados, lo cual indica que efectivamente existe un patrón que el modelo está encontrando y los resultados no se debe a la aleatoriedad intrínseca del algoritmo de simulación.

Calibración de los parámetros

Aunque se podría pensar que fijar $M,\ J$ y K a discreción del estadístico sesga los resultados, en realidad es sólo una consecuencia de haber escogido un modelo estructurado como se hizo. Prácticamente en ningún modelo estadístico, incluso en los no paramétricos, se puede abandonar toda decisión al algoritmo para que este encuentre el modelo perfecto. Siempre existirá un parámetro o variable que se deba de afinar lo cual introduce una dimensión subjetiva al modelo, **wasserman2007all**. Inclusive, la misma selección del modelo introduce variabilidad no sistemática en los datos lo cual podría sesgar los resultados. Sin embargo, en el caso de los parámetros del modelo, existe un proceso de calibración que se puede realizar de forma analítica y no por fuerza bruta. Siempre se busca entender el porqué esa selección puntual de parámetros no funciona y así, modificar el modelo en respuesta. En particular para los ejemplos presentados la selección de $M,\ J$ y K, para los casos sencillos, era prácticamente trivial y el modelo lograba capturar el patrón con diferentes tipos de fronteras. Como ejemplo, se tienen las realizaciones del primer ejemplo de la sección **??**.

Velocidad del algoritmo

Es curioso notar, que aunque el modelo sea complejo y pueda crecer rápidamente en el número de parámetros λ al modificar $M,\ J$ y K o aumentar el número de covariables d, la velocidad del algoritmo es relativamente buena. Gracias a las opti-

mizaciones realizadas en los cálculos parciales y el uso de distribuciones conjugadas, la simulación de un gran número de parámetros es trivial. Prácticamente en todos los modelos probados, se terminaron de simular las cadenas en un minuto o menos. Aquello que hace que el algoritmo sea más lento, usualmente es escalar n ordenes de magnitud. Otro factor importante que influye en la velocidad del algoritmo es el uso de un paradigma bayesiano para el entrenamiento. Esta decisión se toma, además de la definición de **albert1993bayesian**, 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 para este modelo, habría logrado que el algoritmo fuera casi instantáneo para un número grande de parámetros.

Gracias a la fácil disponibilidad y uso 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, tanto el modelo como el algoritmo se puede ir mejorando con contribuciones de terceros. Dado que el algoritmo se implementó en el software estadístico `R`, el cual tiene múltiples ventajas, se reconoce que no es el lenguaje más veloz computacionalmente pues corre a un nivel muy alto y es un lenguaje interpretado. Si se pensara usar el algoritmo para aplicaciones más robustas, se recomendaría adaptar las funciones a un lenguajes de nivel más bajo como lo puede ser `C++`.

0.2. Posibles mejoras y actualizaciones

Como se mencionó, la fuerza del modelo recae en la combinación de todos sus componentes pues le otorga flexibilidad a las estructuras contenidas en η. No obstante, como se estudió en el ejemplo 5, el modelo tiene limitantes que se pueden mejorar.

La primer y más urgente mejora que se propone explorar, es la de incorporación de un método para la selección de covariables. Bajo el enfoque de la estadística frecuentista para modelos de regresión, en ocasiones se trata de buscar aquellas covariables más significativas para la predicción correcta 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, **bishop2006pattern**.[1] Los métodos de aprendizaje de máquina más recientes son especialmente efectivos en este ámbito; sus algoritmos recaen en usar cantidades enormes de información con múltiples covariables ($d >> 1$) para hacer predicciones robustas al entrenar miles de parámetros, **nielsen2015neural**. Bajo el paradigma bayesiano la selección de covariables también se puede manejar. Los métodos más usados, incorporan otra serie nueva serie de variables auxiliares (usualmente funciones indicadoras) cuyo trabajo es detectar cuando una variable es relevante o no. A estas variables, también se les da un tratamiento bayesiano y son estimadas por los mismos algoritmos MCMC a la par de todos los demás, **o2009review**.

1. Usualmente el criterio de Akaike

Para este trabajo, como se observa en el ejemplo 6, la selección de covariables se hizo de manera manual (y subjetiva) 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 covariables, no sólo se realiza en afán de simplificar los modelos, sino por una razón computacional de convergencia pues al no tener covariables adicionales los parámetros asociados a las variables relevantes serían más significativos. Asimismo, se podría reducir la colinealidad entre covariables.

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 más 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 de muestreo pues la dimensión del número de parámetros fluctúa como se agregan o se eliminan nodos. 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.

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 (**??**). Esto permite que el muestreo sea sencillo, requiriendo únicamente álgebra lineal y simulaciones de variables con distribución normal multivariada. Aunque el supuesto de que $\boldsymbol{\beta} \sim \mathcal{N}_\lambda$ 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, requeriría de cambiar sustancialmente el algoritmo, y por ende las derivaciones. Asimismo, se estaría obligando a usar paquetes de software que permitan hacer inferencia bayesiana más general como las librerías `STAN` o `BUGGS`.

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 funciona muy bien, sin embargo, es computacionalmente lenta. Por ejemplo, si se incorporara el cambio en la posición de los nodos sería forzoso recalcular la matriz $\widetilde{\Psi}$ múltiples veces. Haciendo un cambio de bases, se puede usar un conjunto de b-splines que representen exactamente el mismo polinomio pero se calculen más rápido. Asimismo, esta modificación permitiría incorporar *splines naturales* que son menos globales y no fluctúan tan rápido más allá de la frontera, **wahba1990splines**.

Estas capacidades adicionales, robustecerían en gran forma al modelo. Si se pensara en usarlo para aplicaciones a gran escala, con miles de datos y aplicaciones concretas, sería importante incorporarlas. Sin embargo, para efectos de este trabajo y para problemas menos trascendentes de clasificación binaria, estas consideraciones se puede obviar. Asimismo, para todos los ejemplos y bases de datos probadas, no

resultaron en un contratiempo.

0.3. El aprendizaje de una máquina

El mundo de la estadística computacional ha sido revolucionado en las últimas décadas gracias a los grandes estadísticos, entre ellos los citados, que han expandido sobre los métodos tradicionales. 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, por ejemplo **madan2015automated** y **shah2014bayesian**. Con este trabajo, además de desarrollar el modelo, se busca sembrar una base teórica y técnica de las posibles extensiones del aprendizaje de máquina, disciplina la cual no es más que estadística computacional llevada al límite.

Algunos de los métodos de aprendizaje de máquina, no son más que extensiones de modelos GLM como el presentado, que se corre un gran número de veces sobre bases de datos con miles de observaciones, 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 de activación, después de colapsar dimensiones, rebasa cierto umbral. Este proceso se repite miles de veces logrando detectar patrones cada vez más complejos. Al final, fuera de las capacidades de estos modelos y su complejidad, la gran mayoría, son regresiones

aumentadas que se basan en los mismos principios que el modelo presentado en este trabajo. Por lo mismo, valía hacer una exploración a fondo de uno modelo análogo y de autoria propia.

La fuerza que han adquirido los métodos de aprendizaje de máquina en los últimos años se debe a que han logrado romper con muchos de los paradigmas tradicionales. Esto pues se han comenzado a aplicar a datos poco tradicionales como lo podrían ser imágenes, textos y sonidos; asimismo, han extendido las capacidades de predicción a un número enorme de categorías y no solo el caso binario. Su utilidad es tan grande que dispositivos de de uso diario, utilizan estos métodos y modelos para clasificar fotos, recopilar información o entender el lenguaje hablado. Sin embargo, vale la pena recordar que *el que una computadora aprenda* es básicamente encontrar patrones, usualmente complejos y no lineales, codificados mediante datos que expresan algún fenómeno de la realidad. Al final, los modelos estadísticos han sido, y siguen siendo, clave para el desarrollo de la ciencia y la tecnología. Por ello, se considera que es más vital que nunca poder entenderlos y analizarlos de forma correcta y bien fundamentada.