Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

BayCon: Generador Bayesiano de Contrafácticos para Inteligencia Artificial Explicable

Tesis de Licenciatura en Ciencias de la Computación

Piotr Romashov

Directora: Maria Vanina Martinez - UBA, Argentina
Codirectores: Marc Langheinreich, Martin Gjoreski - USI, Suiza
Buenos Aires, 2022

BAYCON: GENERADOR BAYESIANO DE CONTRAFÁCTICOS PARA INTELIGENCIA ARTIFICIAL EXPLICABLE

Generar contrafácticos para descubrir escenarios predictivos hipotéticos es el estándar de facto para explicar los modelos de aprendizaje automático y sus predicciones. Sin embargo, construir un explicador contrafáctico que sea eficiente en el tiempo, escalable y agnóstico del modelo, además de ser compatible con atributos continuos y categóricos, sigue siendo un desafío abierto.
Para complicar aún más las cosas, garantizar que las instancias a contrastar están optimizadas para la esparsitud de los atributos, permanecen cerca de la instancia explicada y se mantiene dentro de la variedad de los datos, está lejos de lo trivial.
Para abordar esta brecha, proponemos *BayCon*: un novedoso generador de contrafácticos basado en el muestreo de características probabilísticas y optimización bayesiana. Tal enfoque puede combinar múltiples objetivos empleando un modelo sustituto para guiar la búsqueda contrafáctica.
Demostramos las ventajas de nuestro método a través de una colección de experimentos basados en seis conjuntos de datos de la vida real que representan tres tareas de regresión y tres de clasificación.

Palabras claves: IA, XAI, Contrafácticos, Bayes, Explicabilidad.

BAYCON: MODEL-AGNOSTIC BAYESIAN CONTERFACTUAL GENERATOR FOR EXPLAINABLE AI

Generating counterfactuals to discover hypothetical predictive scenarios is the de facto standard for explaining machine learning models and their predictions. However, building a counterfactual explainer that is time-efficient, scalable, and model-agnostic, in addition to being compatible with continuous and categorical attributes, remains an open challenge.
To complicate matters even more, ensuring that the contrastive instances are optimised for feature sparsity, remain close to the explained instance, and are not drawn from outside of the data manifold, is far from trivial.
To address this gap we developed a novel counterfactual generator based on probabilistic feature sampling and Bayesian optimisation. Such an approach can combine multiple objectives by employing a surrogate model to guide the counterfactual search.
In this work we demonstrate the advantages of our method through a collection of experiments based on six real-life datasets representing three regression tasks and three classification tasks.

Keywords: IA, XAI, Counterfactuals, Bayes, Explainability.

AGRADECIMIENTOS

Es irónico que se le llame carrera a la carrera de grado, cuando la clave es la constancia y el esfuerzo, al contrario la rapidez solo genera ansiedad y no disfrutar el proceso. El recorrido fue largo, formándome no solo en computación, sino como persona, y la vida pasó a su lado también, y con ella muchas personas que me acompañaron en esta travesía, que hicieron posible que sea ameno y disfrutable todo este período, tanto adentro como afuera de la universidad. En los momentos donde era tan difícil, crisis de dejar las materias por la mitad, o incluso dejar los estudios por completo, me ayudaron y supieron alentar e incentivar a seguir el camino. Y acá estamos, 10 años luego de comenzar, terminando, y comenzando nuevos caminos. Pero sin dudas, con alegría de terminar, sintiéndome afortunado por todas las personas que me acompañaron, las amistades, y las experiencias vividas. Gracias.

Cuando arranqué a estudiar me contaron que la tesis en una carrera de ciencias exactas es ampliar la periferia de la ciencia, aportar un nuevo grano de arena que haga avanzar el conocimiento humano. Pocas veces en mi tiempo estudiando tuve posibilidades de dedicarme exclusivamente a un proyecto, pero surgió la oportunidad de una beca de investigación en una universidad en Suiza, la *Università della Svizzera italiana*, y realizar mi trabajo de tesis allá. Gracias a la USI, a mi directora acá, y a mis supervisores allá. Fue una experiencia maravillosa y me llena de alegría de finalmente poder realizar este aporte de mi granito de arena.

A la maravillosa comunidad de la UBA, de exactas, y en particular de computación, que hace que todo esto sea posible, poniéndole todo el esfuerzo que requiere el proveerle oportunidades a formarse a todo el que quiera, a tener más oportunidades en la vida, gracias de corazón.

Y a tantos amigos en el camino de la vida, que lo hicieron tanto más disfrutable. A todos ustedes, muchas gracias y a seguir recorriendo juntos!

Índice general

1. INTRODUCCIÓN

Vivimos en una revolución tecnológica, parecida a la revolución industrial, en la que buscamos automatizar y delegar tareas cada vez más complejas a la computadora, estas tareas abarcan un gran espectro, que pueden ir desde detectar una sonrisa o una cara en particular en una imagen, hasta el análisis de reincidencia en actividad criminal, sentencias y el otorgamiento o no de un crédito económico.

Durante las primeras décadas del siglo 21 ha habido un creciente interés e inversiones en la inteligencia artificial. Sobre todo cuando el aprendizaje automático (machine learning - ML) ha sido exitosamente aplicado a muchos problemas en la academia e industria, entre muchos, nos ha dado los autos autónomos, reconocimiento del habla, y amplio considerablemente nuestro entendimiento del genoma humano. Todo esto debido a nuevos métodos, mayor poder de procesamiento, y el surgimiento de las inmensas colecciones de datos.

Los algoritmos de IA se volvieron tan complejos que si quisiéramos observar lo que sucede ahí adentro, resulta inentendible siquiera para sus creadores, ya que la computadora internamente desarrolló su propio lenguaje (muchísimo mas eficiente que el humano para la tarea a realizar), y que es solo comprensible para ella misma. Imaginemos por un momento matrices (como tablas de excel), que tienen miles, quizás millones de columnas y filas con información, y ni siquiera los programadores, o cualquier otro observador, entiende exactamente cómo está operando. Algo así como saber que estamos pensando ahora mismo con tan solo por mirar una radiografía de nuestro cerebro. Ya no estamos programando, estamos creando inteligencia, la cual se hace mas difícil poder interpretar los modelos a medida que se complejizan.

Estos son los mecanismos en funcionamiento detrás de sistemas tan populares como Facebook, Instagram, Youtube, Google, Amazon, etc. Para dar un ejemplo, a cuantos nos ocurrió de entrar a Youtube a mirar un vídeo de 5 minutos, y terminar mirando 27 vídeos en dos horas? Esto es debido a que analiza nuestros gustos y estado, y nos recomienda el contenido más probable que nos vaya a gustar. Podríamos decir que hasta ahora no hay nada malo en todo esto, pero ¿Qué pasa si el sistema decide que es más rentable venderle boletos a Las Vegas a gente con trastorno bipolar y que están entrando en un estado maniaco? Dichas personas tienden a ser más gastadores, y jugadores compulsivos. Estos sistemas podrían hacerlo, e incluso sus desarrolladores no tener pista que están haciendo recomendaciones que son de, por lo menos, dudosa moral.

Los sistemas de IA son cada vez más sofisticados y en muchos casos una “caja negra”. Cuando las decisiones derivadas de tales sistemas afectan vidas humanas, directa o indirectamente, como lo puede ser en medicina, leyes o defensa, hay una necesidad emergente de entender como dichas decisiones son realizadas por los métodos de IA. [1][17]

1.1. Inteligencia Artificial - IA

Es común imaginarse a la inteligencia artificial en forma de robot humanoide, como nos lo presentó Hollywood en sus películas como; "El hombre bicentenario" o "Yo robot" (grandes creaciones de Isaac Asimov, entre muchas otras), pero la realidad es que no hace falta que sea tangible para que pueda afectarnos, sobre todo, porque lo que hace que algo sea inteligente, no es que tenga extremidades, o similitud física con un humano, sino lo que importa es la capacidad de interpretar correctamente datos externos, para aprender de dichos datos y emplear esos conocimientos para lograr tareas y metas concretas a través de adaptación, en otras palabras; resolver problemas y aprender a resolverlos mejor en el proceso.

La IA, con mayores o menores capacidades de resolución de problemas, es cada vez mas ubicua, se encuentra por ejemplo en los celulares de la mayoría de las personas, que incluyen la detección de objetos en fotos, distinguir caras y voces con personas, o el sentimiento de una conversación.

1.2. IA basada en datos

La concepción de que es inteligencia artificial fue cambiando a medida que fueron desarrollándose técnicas y tecnologías cada vez mas avanzadas, algoritmos de ultima generación que fueron catalogados como IA, quedaron relegados a algoritmos de programación tradicional. Hoy en día cuando se habla de inteligencia artificial, generalmente es una de sus sub-áreas; aprendizaje automático (Machine Learning en inglés - ML), o algunas de sus ramas más novedosas como aprendizaje profundo (Deep Learning en inglés - DL). Estas son técnicas de algoritmos que aprenden patrones relevantes a partir de cantidades de datos gigantescas, con los que van aprendiendo y corrigiendo su aprendizaje para mejorar los resultados.

Las inteligencias artificiales se encuentran en el núcleo de muchos sectores y rubros que han incorporado las nuevas tecnologías de la información. Hay un claro consenso en la extrema importancia de maquinas inteligentes dotadas con habilidades de aprendizaje, razonamiento y adaptación. Es por la virtud de estas capacidades que los métodos de IA están logrando niveles de rendimiento sin precedentes al aprender a resolver tareas computacionales cada vez mas complejas, haciéndolos pivotales para el desarrollo futuro de la sociedad humana. [1][54].

1.2.1. Machine Learning - Aprendizaje automático

El aprendizaje automático se refiere a enseñar a una maquina a aprender de los datos y cambiar cuando se encuentra expuesto a nuevos datos. Formalmente, el aprendizaje automático es un tipo de inteligencia artificial que provee a las computadoras con la habilidad de aprender de ejemplos o experiencia sin ser explícitamente programados.

ML ha demostrado ser valioso porque puede resolver problemas a una velocidad y escala que la mente humana no puede duplicar por sí sola. [46];

- *Los datos son clave*: los algoritmos que impulsan el aprendizaje automático son fundamentales para el éxito. Los algoritmos de ML construyen un modelo matemático

basado en datos de muestra, conocidos como "datos de entrenamiento", para hacer predicciones o tomar decisiones sin estar programados explícitamente para hacerlo. Esto puede revelar tendencias dentro de los datos que las empresas de información pueden usar para mejorar la toma de decisiones, optimizar la eficiencia y capturar datos procesables a escala.

- *IA es el objetivo*: ML proporciona la base para los sistemas de IA que automatizan procesos y resuelven problemas comerciales basados en datos de manera autónoma. Permite a las empresas reemplazar o aumentar ciertas capacidades humanas. Las aplicaciones comunes de aprendizaje automático que puede encontrar en el mundo real incluyen chatbots, autos sin conductor y reconocimiento de voz.

El aprendizaje automático no es ciencia ficción. Ya es ampliamente utilizado por empresas de todos los sectores para promover la innovación y aumentar la eficiencia de los procesos [46];

Seguridad Los modelos de aprendizaje automático pueden identificar vulnerabilidades de seguridad de datos antes de que se conviertan en infracciones. Al observar experiencias pasadas, los modelos de aprendizaje automático pueden predecir futuras actividades de alto riesgo para que el riesgo pueda mitigarse de manera proactiva.

Finanza Los bancos, las casas de bolsa y las empresas fintech utilizan algoritmos de aprendizaje automático para automatizar las operaciones y brindar servicios de asesoramiento financiero a los inversores. Bank of America está utilizando un chatbot, Erica, para automatizar la atención al cliente.

Salud ML se utiliza para analizar conjuntos masivos de datos de atención médica para acelerar el descubrimiento de tratamientos y curas, mejorar xlos resultados de los pacientes y automatizar procesos de rutina para evitar errores humanos. Por ejemplo, Watson de IBM utiliza la minería de datos para proporcionar a los médicos datos que pueden usar para personalizar el tratamiento del paciente.

Fraude La IA se está utilizando en el sector financiero y bancario para analizar de forma autónoma un gran número de transacciones para descubrir actividades fraudulentas en tiempo real. La firma de servicios tecnológicos Capgemini afirma que los sistemas de detección de fraude que utilizan aprendizaje automático y análisis minimizan el tiempo de investigación de fraude en un 70 % y mejoran la precisión de detección en un 90 %.

Venta Los investigadores y desarrolladores de IA están utilizando algoritmos de ML para desarrollar motores de recomendación de IA que ofrecen sugerencias de productos relevantes basadas en las elecciones anteriores de los compradores, así como en datos históricos, geográficos y demográficos.

1.2.2. Machine Learning vs Programación tradicional

Para explicarlo, usaremos un ejemplo concreto de estimación del riesgo de crédito [10]; Supongamos que nuestro objetivo es evaluar automáticamente el riesgo de que un potencial prestatario pague o no un préstamo. La información a la que podemos acceder incluye el perfil de los solicitantes de préstamos (nombre, sexo, edad, profesión, estado civil, ingresos, ahorro, historial financiero anterior, etc.) y registros de préstamos anteriores que

contienen el perfil de los prestatarios anteriores y si finalmente pagaron los préstamos o no. Entonces, si usamos un enfoque tradicional para hacer una predicción, la entrada a la máquina son datos y programa. Aquí, los datos son la información de un nuevo solicitante y el programa es un conjunto de reglas que pueden definirse según las experiencias pasadas de los gerentes (por ejemplo, si los ingresos son > 150 000 Y el ahorro es > 200 000, entonces Pagar = Sí). Según las reglas y el perfil, la máquina mostrará si el prestatario pagará su préstamo o no.

Sin embargo, para el enfoque de aprendizaje automático, los datos se denominan **instancias de datos** de entrada de los registros históricos de prestamos anteriores, y la salida aquí se refiere al valor de la columna de salida: resultado final, ya sea que el prestatario pague su préstamo o no. Usando los datos históricos, un algoritmo aprende la relación entre los datos de entrada y la salida, que es el llamado **modelo de predicción**. Este modelo de predicción se puede considerar como el mapeo de los datos de entrada a la salida. Una vez que se aprende el modelo, dado un perfil de información del nuevo prestatario, el modelo automáticamente hará una predicción [10].

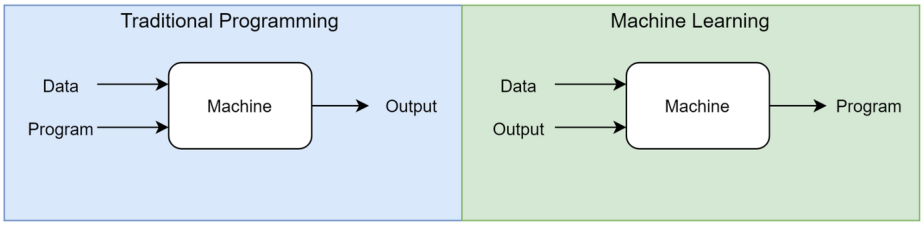


Fig. 1.1: Programación tradicional vs. Aprendizaje automático [10]

Ahora sabemos que la característica del aprendizaje automático es intentar aprender un modelo a partir de datos . Por lo tanto, el aprendizaje automático es un enfoque basado en datos en lugar de pedirle a un experto que proporcione reglas hechas a mano. Además, el aprendizaje automático se centra en el desarrollo de un algoritmo general para aprender un modelo a partir de datos en lugar de centrarse en problemas de aplicación específicos. Para diseñar un algoritmo general, necesitamos usar muchas matemáticas y estadísticas aplicadas para modelar el problema y también usar técnicas informáticas para hacer que el algoritmo sea más eficiente.

Actualmente, algunos investigadores propusieron utilizar algunos hallazgos de la ciencia cognitiva para diseñar el algoritmo de aprendizaje automático, por ejemplo, el aprendizaje profundo (Deep Learning) o la red neuronal (Deep Neural Networks) es uno de los hallazgos de la ciencia cognitiva y está motivado por el cerebro humano. En el aprendizaje automático, existen muchas metodologías diferentes, como el aprendizaje profundo, los modelos de probabilidad, los métodos kernel, los métodos basados en la entropía, etc [10].

1.2.3. Limites

Es importante comprender lo que el aprendizaje automático puede y no puede hacer. Tan útil como lo es la automatización de la transferencia de inteligencia humana a las máquinas, está lejos de ser una solución perfecta para sus problemas relacionados con los datos. Entre algunos de los aspectos limitantes, se encuentran [46]:

- El aprendizaje automático no se basa en el conocimiento. Contrariamente a la creencia popular, el aprendizaje automático no puede alcanzar la inteligencia a nivel humano. Las máquinas son impulsadas por datos, no por el conocimiento humano. Como resultado, la "inteligencia" está dictada por el volumen y la calidad de datos con los que se entrena.

- Los modelos de aprendizaje automático son difíciles de entrenar. La mayoría de los científicos de datos admite que entrenar IA con datos es más difícil de lo esperado. Se necesita tiempo y recursos para entrenar máquinas. Se necesitan conjuntos de datos masivos para crear modelos, y el proceso implica el etiquetado previo manual y la categorización del conjuntos de datos. Este drenaje de recursos puede crear latencia y cuellos de botella en el avance de las iniciativas de ML.

- El aprendizaje automático es propenso a problemas de datos. Casi todas las empresas han experimentado problemas relacionados con el entrenamiento, con la calidad de los datos, el etiquetado de datos y la confianza en el modelo de construcción. Esos problemas relacionados con el entrenamiento son una razón clave por la que gran parte de los proyectos de ML se estancan antes de la implementación. Esto ha creado un umbral extraordinariamente alto para el éxito de ML.

- El aprendizaje automático a menudo está sesgado y en general son poco interpretables. Esto implica que ante la presencia de sesgo, puede ser muy difícil, sino imposible, identificar la fuente y corregir la toma de decisiones sesgada. Es importante destacar que reentrenar el modelo puede no ser suficiente si la fuente del sesgo no puede ser identificada y administrada.

1.2.4. Desafíos

Dentro de los principales desafíos que se fue enfrentando la comunidad de IA, la mayoría de la literatura apunta hacia hacer modelos subyacentes que sean cada vez mas rápidos, ya sea en el entrenamiento o en la predicción, y mas precisos. Esto llevo a que muchos de los modelos tomen complejidades cada vez mayores, en los cuales los mismos diseñadores carezcan de entendimiento sobre que esta pasando internamente y el porque de una predicción.
Mientras que los modelos "caja negra" de ML están siendo incrementalmente usados para hacer predicciones importantes en contextos críticos, la demanda por transparencia esta incrementando también de los diferentes interesados en IA [1][41]. El peligro esta en crear y usar decisiones que no son justificables, legitimas, o que simplemente no permite obtener explicaciones detalladas de su comportamiento [19]. Explicaciones que apoyan el resultado de un modelo son cruciales para una amplia variedad de dominios, por ejemplo la medicina de precisión, donde los expertos requieren mucha mas información del modelo que una simple predicción binaria para apoyarse en su diagnostico [50] donde una predicción incorrecta suele acarrear un alto costo. Otros ejemplos son el de los vehículos autónomos en transporte, seguridad, y finanzas, entre otros.

1.3. Explicabilidad e IA

A medida que los sistemas de IA son incrementalmente desplegados en la vida cotidiana, a habido un incremento también en artículos de investigación en el problema de

eXplainable AI (XAI), impulsado por la preocupación por si estos sistemas son justos, responsables y confiables [28]. Un claro ejemplo de esto es dentro del dominio de medicina, un médico puede basarse en una inteligencia artificial para ayudar a realizar determinados diagnósticos, como el de si una biopsia contiene o no células cancerígenas. En este caso la predicción carece de valor si no viene acompañada por una explicación que lo justifique, dado que el error de predecir algo incorrecto puede acarrear graves consecuencias humanas, el médico debería poder verificar que el modelo este funcionando bien, un ejemplo de explicación es a través de explicaciones visuales; generar una imagen con un mapa de calor que indique donde es que se detecto en la célula un patrón cancerígeno.

El "derecho a la explicación" sugerido por el Reglamento de Protección General de Datos de la Comisión Europea (GDPR) [17] desafió a la comunidad de aprendizaje automático a construir explicabilidad en modelos predictivos y sus resultados. Este cambio de paradigma, donde el rendimiento predictivo ya no deja de ser el único (y principal) objetivo - da lugar a dos puntos de vista distintos;

- Uno argumenta que las cajas negras algorítmicas deben continuar siendo optimizadas para el poder predictivo con necesidades de explicabilidad, posiblemente satisfechas a través de métodos post-hoc debido a una aparente incompatibilidad de estos dos objetivos, obligando así a uno de ellos a ser sacrificado por el otro.

- El segundo punto de vista cuestiona esta compensación como puramente anecdótica, y argumenta persuasivamente a favor de la construcción inherentemente transparente Modelos ML, especialmente para decisiones de alto riesgo.

De acuerdo a GDPR, todos los modelos de aprendizaje automático deberían ofrecer la posibilidad de responder/proveer explicaciones del estilo: *"Le negaron un préstamo porque su ingreso anual era de $30,000. Si sus ingresos hubieran sido de $45.000, le habrían ofrecido un préstamo"* [53]. Este ejemplo nos introduce a un estilo particular de explicaciones, las *explicaciones mediante contrafácticos*.

Es interesante notar la necesidad latente de modelos interpretables de inteligencia artificial IA lo largo del tiempo (lo que se ajusta a la intuición, ya que la interpretabilidad es un requisito en muchos escenarios), pero no ha sido hasta 2017 cuando el interés por las técnicas para explicar los modelos de IA se han extendido a la comunidad académica (Figura 1.2) [1].

Antes de continuar, es conveniente establecer primero un punto común de entendimiento sobre qué significa el término explicabilidad en el contexto de la IA y, más específicamente, de ML. Una de las cuestiones que dificulta el establecimiento de puntos comunes es el uso intercambiable de interpretabilidad y explicabilidad en la literatura. Hay diferencias notables entre estos conceptos. Para resumir la nomenclatura más utilizada, en esta sección aclaramos la distinción y las similitudes entre los términos que se usan a menudo en el comunidades éticas de IA y XAI. [1]

- En general, **interpretabilidad** se refiere a una característica pasiva de un modelo, refiriéndose al nivel en el que un modelo dado tiene sentido para un ser humano observador. Esta característica también se expresa como **transparencia**.

- Por el contrario, la **explicabilidad** puede verse como una característica activa de

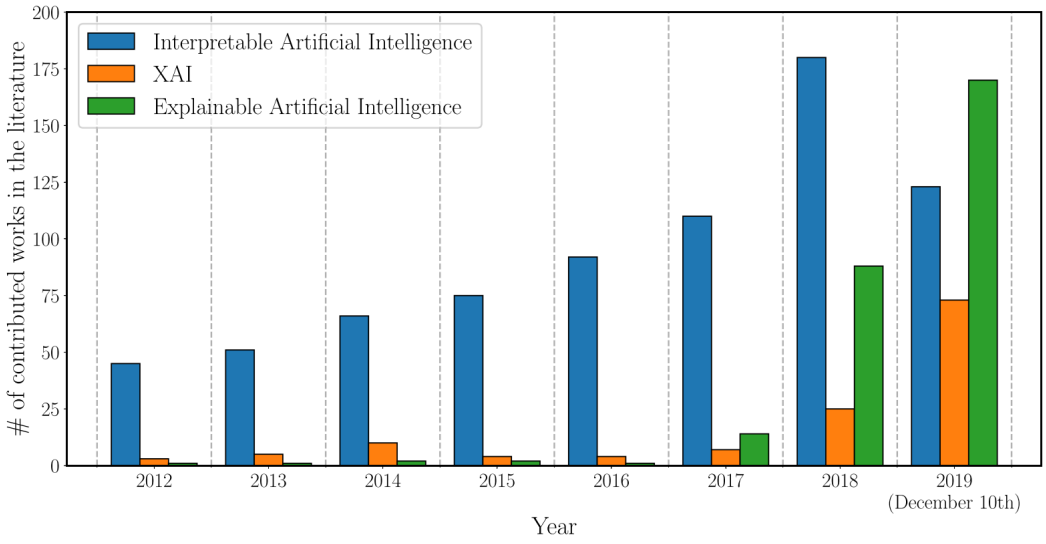


Fig. 1.2: Evolución del número de publicaciones totales cuyo título, resumen y/o palabras clave hacen referencia al ámbito de la XAI durante los últimos años. Datos recuperados de Scopus® (10 de diciembre de 2019) utilizando los términos de búsqueda indicados en la leyenda al consultar esta base de datos. [1]

un modelo, que denota cualquier acción o procedimiento realizado por un modelo con la intención de aclarando o detallando sus funciones internas.

- *Inteligibilidad*: denota la característica de un modelo para hacer que un ser humano entienda su función – cómo funciona el modelo, sin necesidad de explicar su funcionamiento interno estructura o los medios algorítmicos por los cuales el modelo procesa datos internamente [37].

- *Comprensibilidad*: En el contexto de modelos ML, la comprensibilidad se refiere a la capacidad de un algoritmo de aprendizaje para representar su conocimiento aprendido de una manera humanamente comprensible [13][15][7]

- *Interpretabilidad*: Es definida como la habilidad de explicar o proveer el significado en medios entendibles para un humano.

- *Explicabilidad*: La explicabilidad está asociada con la noción de explicación, como una interfaz entre los humanos y un tomador de decisiones, es decir que al mismo tiempo, es tanto un proxy preciso del tomador de decisiones como comprensible para los humanos [18].

- *Transparencia*: Un modelo se considera transparente si por sí mismo es entendible.

1.3.1. Fundaciones filosóficas - ¿Que es una explicación?

Para analizar que es una explicación, tomemos parte del trabajo de Miller en *Explanation in Artificial Intelligence: Insights from the Social Sciences* [35] , en su estudio cita al trabajo de Lewis [31], con la definición la explicación;

> *Explicar un evento es proporcionar alguna información sobre su historia causal. En un acto de explicación, alguien que está en posesión de alguna información sobre la historia causal de algún evento (la llamaremos información explicativa) trata de transmitirla a otra persona.* [31]

¿Porque las personas piden explicaciones? Hay muchas razones por las que la gente puede pedir explicaciones. La curiosidad es el criterio primario que usan los humanos, pero otras razones pragmáticas incluyen el examen, (e.g.; una maestra que pide a sus alumnos una explicación sobre un examen con el propósito de probar el conocimiento de los estudiantes sobre un tema en particular, y explicación científica; preguntando por qué observamos un fenómeno ambiental en particular). En este trabajo, estamos interesados en la explicación en IA y, por lo tanto, nuestro enfoque está en cómo los agentes inteligentes pueden explicar sus decisiones. Como tal, esta sección se ocupa principalmente por qué la gente pide explicaciones "cotidianas" de por qué ocurren eventos específicos, en lugar de explicaciones para fenómenos científicos generales, aunque este trabajo sigue siendo relevante en muchos casos. [35]

Está claro que la función principal de la explicación es facilitar el aprendizaje [33][56]. A través del aprendizaje, obtenemos mejores modelos de cómo se producen eventos o propiedades particulares, y somos capaces de utilizar estos modelos a nuestro favor. Heider [20] afirma que la gente busca explicaciones para mejorar su comprensión de alguien o algo para que puedan puede derivar un modelo estable que se puede utilizar para la predicción y el control. Esta hipótesis está respaldado por investigaciones que sugieren que las personas tienden a hacer preguntas sobre eventos o observaciones que consideran anormales o inesperadas desde su propio punto de vista [23][22][21].

Malle [34], quien ofrece quizás la discusión más completa de explicaciones en la vida cotidiana, en el contexto de explicar la acción/interacción social, argumenta que las personas piden explicaciones por dos razones:

- Encontrar significado: reconciliar las contradicciones o inconsistencias entre elementos de nuestras estructuras de conocimiento.
- Gestionar la interacción social: crear un significado compartido de algo y cambiar las creencias e impresiones de los demás, sus emociones o influir en sus acciones.

Crear un significado compartido es importante para la explicación en IA. En muchos casos, una explicación proporcionada por un agente inteligente será precisamente para hacer eso - para crear un comprensión compartida de la decisión que se tomó entre él y un observador humano, al menos hasta cierto nivel parcial.

Según Miller [35], uno de los hallazgos más importantes en la literatura filosófica y de ciencias cognitivas desde la perspectiva de la IA explicable: es el de la *explicación contrastiva*, o también llamada *explicación contrafáctica*. Las investigaciones muestran que las personas no explican las causas de un evento per se, sino explican la causa de un evento en relación con algún otro evento que no ocurrió; es decir, una explicación es siempre de la forma ¿Por qué P en vez de Q?, en la que P es el evento objetivo y Q es un caso de contraste contrafáctico que no ocurrió, incluso si Q está implícita en la pregunta. Esto se llama explicación contrastiva. Algunos autores se refieren a Q como el caso contrafáctico [33][21][23].

Pregunta	Razonamiento	Descripción
¿Qué?	Asociativo	Razón por la cual los eventos no observados podrían haber ocurrido dados los eventos observados.
¿Cómo?	Intervencionista	Simule un cambio en la situación para ver si el evento todavía sucede
¿Por qué?	Contrafáctico	Simular causas alternativas para ver si el evento todavía sucede

Tab. 1.1: Clases de preguntas explicativas y el razonamiento requerido para contestar [35]

En esta ultima categoría de razonamientos, los contrafácticos, se enfocara el estudio de este trabajo. Los tipos, niveles, y estructuras de las explicaciones, se pueden encontrar más en profundidad en el estudio de Miller [35].

1.3.2. IA Explicable - XAI

Explainable AI propone crear una serie de técnicas de ML que produzcan modelos explicables, manteniendo el alto nivel de rendimiento del aprendizaje (e.g.; precisión de la predicción), y al mismo tiempo permita a los humanos entender, confiar, y efectivamente manejar la emergente generación de inteligencias artificiales [19].

XAI trae consigo perspectivas de las Ciencias Sociales [35], y considera la psicología de la explicación; antes que nada, veamos primero algunas definiciones que plantea un estudio de XAI [1], en el cual busca responder *¿Que?*, *¿Por qué?*, *¿Para qué?*, y *¿Cómo?*.

¿Que? Para responder esto usaremos una de las contribuciones del estudio en XAI de Arrieta et al. [1] en la definición de explicabilidad dentro del contexto de IA.

> *Dada una audiencia, una Inteligencia Artificial explicable es aquella que produce detalles o razones para que su funcionamiento sea claro o fácil de entender.* [1]

¿Por que? Como se indicó anteriormente, la explicabilidad es una de las principales barreras que enfrenta la IA hoy en día en lo que respecta a su implementación práctica. La incapacidad para explicar o comprender completamente las razones por las cuales los algoritmos de ML de última generación funcionan tan bien como lo hacen, es un problema que encuentra sus raíces en dos causas diferentes [1]:

- La primera causa es la brecha entre la comunidad investigadora y los sectores empresariales, lo que impide la penetración total de los modelos ML más nuevos. En general este problema se presenta en sectores estrictamente regulados con cierta reticencia a implementar técnicas que puedan poner en riesgo sus activos, como banca, finanzas, seguridad y salud, entre muchos otros.
- El segundo eje es el del conocimiento. La IA ha ayudado a la investigación en todo el mundo con la tarea de inferir relaciones que estaban mucho más allá del alcance cognitivo humano. Sin embargo, estamos entrando en una era en la que los resultados y las métricas de rendimiento son el único interés que se muestra en los estudios de investigación. Aunque para ciertas disciplinas este podría ser el caso justo, la ciencia y la sociedad están lejos de estar preocupadas solo por el rendimiento.

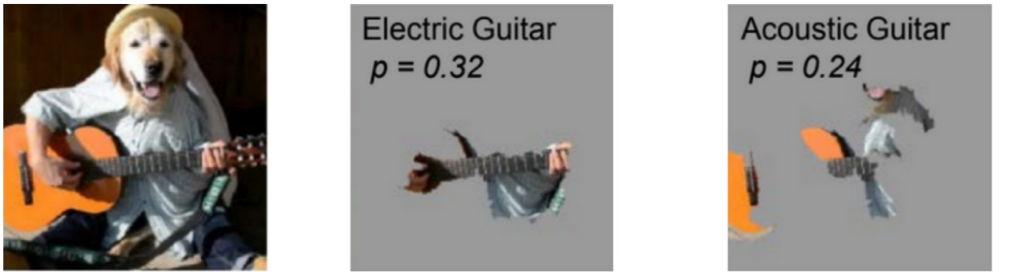
¿Para que? La actividad de investigación en torno a XAI ha expuesto hasta ahora diferentes objetivos a partir del logro de un modelo explicable, sin embargo casi ninguno de los artículos revisados esta completamente de acuerdo con los objetivos requeridos para describir lo que debería cumplir un modelo explicable. En el estudio [1] se sintetizan y enumeran las definiciones de estas metas de XAI, para establecer un primer criterio de clasificación;

- *Integridad*: La integridad puede ser considerada como la confianza de si un modelo actuará según lo previsto cuando se enfrente un problema dado.
- Causalidad: Otro objetivo común de la explicabilidad es encontrar causalidad entre las variables de datos.
- *Transferibilidad*: La cantidad de artículos que afirman que la capacidad de hacer que un modelo sea explicable es comprender mejor los conceptos necesarios para reutilizarlo o mejorar su rendimiento es la segunda razón más utilizada para buscar la explicabilidad del modelo.
- *Informatividad*: Los modelos ML se utilizan con la intención final de apoyar la toma de decisiones [26]. Los modelos de aprendizaje automático explicables deben proporcionar información sobre el problema que se está abordando.
- *Confianza*: Como generalización de robustez y estabilidad, la confianza siempre debe evaluarse en un modelo en el que la confiabilidad se espera.
- *Justicia*: Desde un punto de vista social, la explicabilidad se puede considerar como la capacidad de alcanzar y garantizar la equidad en los modelos de ML. El soporte de algoritmos y modelos está creciendo rápidamente en campos que involucran vidas humanas, por lo tanto, la explicabilidad debe considerarse como un puente para evitar el uso injusto o poco ético de los resultados de los algoritmos.
- Accesibilidad: Permite a los usuarios finales obtener más involucrados en el proceso de mejora y desarrollo de un determinado modelo de ML.
- *Interactividad*: Una vez más, este objetivo está relacionado con campos en el que los usuarios finales son de gran importancia, y su capacidad para ajustar e interactuar con los modelos es lo que asegura el éxito.
- *Conciencia de privacidad*: Uno de los subproductos habilitados por la explicabilidad en los modelos ML es su capacidad para evaluar la privacidad. No ser capaz de entender lo que ha sido capturado por el modelo [6] y almacenado en su representación interna puede implicar una violación de la privacidad.

¿Cómo? La literatura hace una clara distinción entre modelos que son interpretables por diseño y aquellos que pueden ser explicados por medio de técnicas XAI externas. Esta dualidad también podría considerarse como la diferencia entre modelos interpretables y técnicas de interpretabilidad de modelos; una clasificación más aceptada es la de modelos transparentes y explicabilidad post-hoc, respectivamente [1].

La explicabilidad post-hoc apunta a modelos que no son fácilmente interpretables por diseño recurriendo a diversos medios para mejorar su interpretabilidad, como *explicaciones*

de texto, *explicaciones visuales*, *explicaciones locales*, *explicaciones por ejemplo*, *explicaciones por simplificación* y *técnicas de explicaciones de relevancia de atributos*, en este último se encuentran incluidas las *explicaciones contrafácticas* [1], que es donde se concentra el trabajo presentado. Cada una de estas técnicas cubre una de las formas más comunes en que los humanos explican los sistemas y procesos ellos mismos. En la Figura 1.3 se puede ver un ejemplo utilizando la técnica LIME para generar explicaciones visuales.



(a) Original image (b) Explaining *electric guitar* (c) Explaining *acoustic guitar*

Fig. 1.3: Ejemplos de explicación usando LIME en imágenes. [1][42]

1.3.3. XAI: Contrafácticos

Los contrafácticos son modificaciones al dato de entrada, las cuales eventualmente cambian la predicción del modelo. Son un enfoque de explicabilidad únicamente posicionados en este espacio ya que pueden ser generados post-hoc pero ser veraz con respecto al modelo subyacente (es decir, mostrar plena fidelidad). Permiten a los usuarios de ML comprender cuál habría sido el resultado de un modelo predictivo si la instancia datos en cuestión hubiese cambiado de una manera particular. Este tipo de análisis contrafáctico ayuda a los explicados a simular ciertos aspectos del modelo ML, mejorando así su interpretabilidad [24].
Como vimos anteriormente, la *evidencia de psicología y las ciencias cognitivas* sugiere que las personas usan razonamiento contrafáctico a diario para analizar lo que podría haber sucedido si hubieran actuado de manera diferente [5].

Las explicaciones a través de contrafácticos proporcionan información a los usuarios sobre lo que podría hacerse para cambiar la resultado de una decisión automatizada (e.g.: "si su paper fuese más novedoso, se habría aceptado en esta conferencia"). En la literatura científica sobre explicaciones post-hoc, que pretenden justificar las predicciones de un modelo de IA después del hecho, la utilidad de dar explicaciones contrafácticas ha ganado considerable tracción, basada en alegaciones de ventajas técnicas, psicológicas y legales [28] Figura 1.4.

La cantidad de contrafácticos que se pueden generar para explicar cualquier evento (un punto de datos seleccionado) puede ser abrumadora [5]. Un enfoque trivial para el problema de generación de contrafácticos es la aplicación de la fuerza bruta o un algoritmo de búsqueda exhaustiva. Sin embargo, el enfoque de fuerza bruta que evalúa todas las alternativas posibles es computacionalmente exigente, y no es razonable para modelos suficientemente grandes. Por ejemplo, para un modelo de decisión con 20 atributos de entrada y cardinalidad de sus conjuntos de valores (escalas) igual a tres (e.g.: 'bajo',

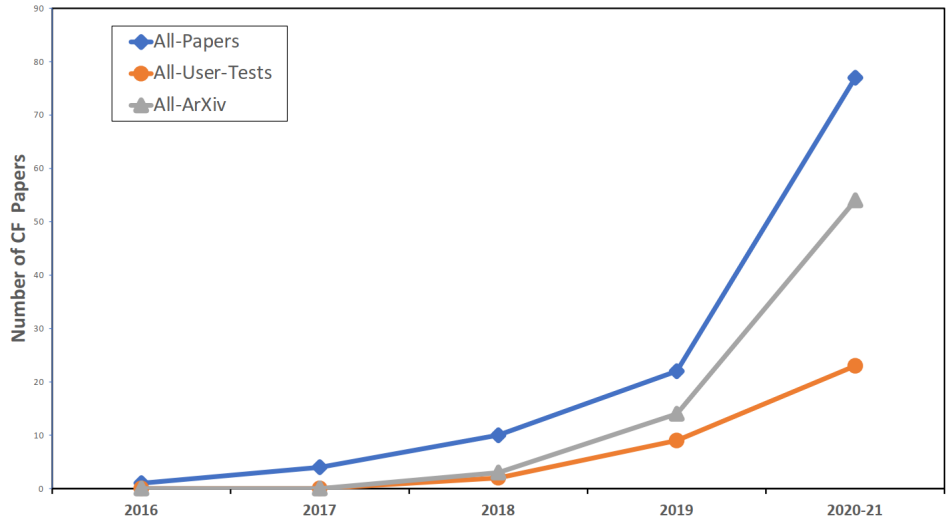


Fig. 1.4: El número de artículos inspeccionados (per annum) sobre counterfactual XAI (2016-2021) sobre (a) métodos CF (All-Papers - azul) (b) estudios de usuarios de CF XAI (All-User-Tests - naranja) (c) ArXiv de una búsqueda de resúmenes usando los términos "contrafactual explanation" (All-ArXiv - gris) [28]

'medio', 'alto'), el número de posibles alternativas (soluciones candidatas) se acerca a los 3.500 millones (3^{20}). En consecuencia, se requieren métodos de búsqueda más avanzados en términos de inteligencia y eficiencia para atravesar el espacio de búsqueda. Dentro de los desafíos a resolver, se encuentran;

- Gran espacio de búsqueda de contrafácticos
- Los métodos que están actualmente disponibles tienden a trabajar para tareas de un solo tipo, o de clasificación, o de regresión.
- Suelen estar restringidos a una familia de modelos específica (por ejemplo, predictores diferenciables)
- Tienen dificultades para lidiar con grandes conjuntos de datos (tanto en el número de instancias como en el de atributos)
- Son computacionalmente ineficientes
- Generan contrafácticos fuera de distribución

2. BAYCON: MODEL-AGNOSTIC BAYESIAN COUNTERFACTUAL GENERATOR

2.1. Introducción

Nuestro trabajo se baso en resolver la problemática de generar explicaciones mediante contrafácticos, es decir, dada una instancia a explicar con su predicción de un modelo de IA, generar instancias con *la menor cantidad de cambios*, tales que modifiquen la predicción del modelo de IA hacia una deseada. El generar contráfacticos lidia con naturaleza combinatoria del conjunto de atributos de entrada y sus dominios (conjunto de valores que cada atributo puede tener), que evoluciona en un complejo problema de optimización combinatoria, donde la dimensionalidad de su espacio de búsqueda afecta la complejidad. El trabajo descripto a continuación fue realizado en el marco de una beca de investigación para estudiantes de máster en la Università della Svizzera italiana (USI - Suiza) — *Research internships for visiting Master students (MaRS programme)*. Los resultados de esta investigación aparecerán publicados en *International Joint Conference on Artificial Intelligence 2022* (IJCAI-22) [43]; este trabajo fue realizado en colaboración con el equipo de investigación de la universidad y referentes del área.

Abordamos estos desafíos con BayCon: un nuevo generador contrafáctico bayesiano, el cual es agnóstico al modelo a explicar. BayCon realiza la búsqueda mediante el uso de la optimización bayesiana, y se ayuda con un modelo sustituto para investigar el equilibrio entre la exploración y explotación de regiones desconocidas y descubrimientos alcanzados, respectivamente. Hasta donde sabemos, es el primer explicador contrafáctico basado en la optimización bayesiana con un modelo sustituto, por lo que es rápido para producir una cantidad considerable de alta calidad instancias contrastivas. Nuestro enfoque es independiente del modelo y compatible con tareas de regresión y clasificación.

El comportamiento y rendimiento de BayCon es analizado con diversas características, incluyendo; tiempo computacional, tiempo para encontrar la primer alternativa apropiada, numero de alternativas (apropiadas) generadas, y numero de cambios en los atributos para alcanzar dichas alternativas. Nuestra evaluación utiliza tres conjuntos de datos de regresión y tres de clasificación con entre 8 y 125 atributos categóricos y numéricos que muestran la velocidad y versatilidad de BayCon. Basado en nuestros experimentos, supera a otros métodos de generación de contrafácticos de última generación, hemos encontrado:

- Mejores soluciones (i.e., menor número de cambios es requerido)
- Eficiente (menor tiempo para generar soluciones)
- Más exhaustivo (mayor cantidad de soluciones)

Este estudio comprende modelos que permiten investigar el rendimiento de BayCon para generar alternativas, pero también brinda una visión clara de la aplicabilidad del BayCon en problemas delimitados por entornos del mundo real. Este tipo de resultados se pueden usar para encontrar los cambios a realizar en nuestros atributos para obtener una predicción deseada, lo cual puede ser muy útil en ámbitos como por ejemplo la agricultura,

donde se dispone de (potencialmente) centenares de atributos y que pueden tomar diversos tipos de valores cada uno. BayCon es una herramienta que puede generar miles de opciones velozmente, con el mínimo cambio en los datos originales del usuario, dándole la posibilidad de elegir cuales esta dispuesto a cambiar, cumpliendo o acercándose al objetivo propuesto por dicho usuario.

2.2. Trabajos Relacionados

Los métodos existentes para generar explicaciones mediante contrafácticos se centran predominantemente en modelos diferenciables aplicados a características continuas [53] [12][38][29]. Esto crea un punto ciego para modelos no diferenciables entrenados en conjuntos de datos con tipos de características mixtas, que son relativamente omnipresentes [44]. Para abordar esta brecha, varios autores propusieron enfoques de programación entera (mixtos) [8][45][27].

Hasta donde sabemos, sólo un estudio propone la optimización bayesiana para la generación de contrafácticos [49]. En ese estudio, el enfoque está solo en los modelos de regresión, y el enfoque propuesto ha sido evaluado solo en dos conjuntos de datos. Además, parece que el manuscrito no está revisado por pares todavía. Además de ese estudio, no pudimos encontrar ningún otro método que utilizan la optimización bayesiana para generar contrafactuales. Por ende, hasta donde sabemos, BayCon es el primer método basado en la optimización bayesiana para generar contrafactuales que funciona tanto para las tareas de clasificación como para las de regresión. Además, según una revisión reciente de la literatura respecto de explicaciones mediante contrafácticos para ML [52], la mayoría de los generadores de contrafácticos existentes funcionan solo con un tipo del modelo (modos lineales, diferenciables o basados en árboles). Además, solo dos (de 29) enfoques independientes del modelo trabajan tanto con atributos numéricos como categóricos. Finalmente, BayCon cumple con las directrices recientes para el diseño de métodos de generación de contrafactuales [28].

Otro método de generación de contrafácticos, que es algo similar a BayCon, es el de "Explicaciones contrafácticas multiobjetivo" (Multi-Objective Counterfactuals - MOC) [9]. MOC es independiente del modelo, compatible con tareas de regresión y clasificación, y capaz de procesar características numéricas y categóricas. Dado que tanto MOC como BayCon intenta abordar el mismo conjunto de carencias de generación de contrafácticos, aunque con enfoques diferentes, los comparamos directamente en un conjunto de experimentos usando seis diversas métricas de evaluación: Tabla 3.2 y Tabla 3.3. Adicionalmente, mostramos cómo BayCon cumple con las directrices recientes para diseñar métodos de generación de contrafácticos, haciéndolo así el enfoque preferido [28].

Bayesian Alternative Generator for Decision Support Models

BayCon se ha basado fuertemente en el trabajo *BAG-DSM Bayesian Alternative Generator for Decision Support Models* [14]. En el presentan un método para abordar el problema de la generación de alternativas para modelos DEX. Más específicamente: dado un modelo jerárquico de atributos múltiples y una alternativa, representando el estado inicial, el objetivo es generar alternativas que exijan el menor cambio en la alternativa

proporcionada para obtener un resultado deseable. BayCon surge como una adaptación de este trabajo, cambiando principalmente su enfoque de modelos cualitativos DEX, hacia los modelos de IA, además usa el análisis previo de los datos de entrenamiento, para luego utilizar en la generación de contráfacticos con atributos que estén dentro de la distribución. Vale la pena ver en detalle la arquitectura de BAG-DSM - Figura 2.1 - ya que BayCon se encuentra basada en ella.

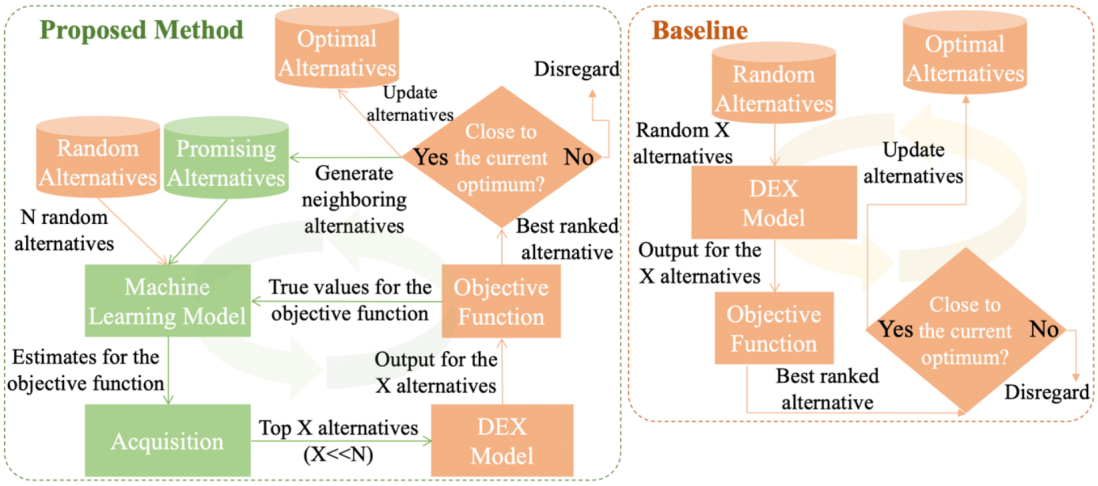


Fig. 2.1: Arquitectura de BAG-DSM (izquierda) vs una ingenua (derecha) [14]

Ambos métodos son un proceso estocástico, y detienen la búsqueda en una cantidad de iteraciones delimitada. Revisemos en detalle la arquitectura de una solución ingenua (derecha Figura 2.1);

1. *Dada una instancia inicial*
2. *Random Alternative*: Generar X alternativas aleatoriamente.
3. *DEX Model*: Obtiene la salida del modelo DEX para las mejores X alternativas y se lo pasa a la función objetiva.
4. *Objective Function*: Con las alternativas y su salida del modelo DEX, (1) revisa si tuvo una salida mejorada (hacia una deseada) y (2) máxima similitud entre la alternativa actual y la nueva propuesta alternativa
5. *Close to the current optimum?*: Filtra las alternativas que se encuentran más distanciadas de la óptima hasta este momento.
6. *Iterar*: Se actualizan las alternativas óptimas hasta el momento y se vuelve a 2.

Ahora notemos las diferencias agregadas del proceso de BAG-DSM (izquierda Figura 2.1);

1. *Dada una instancia inicial*
2. *Promising Alternatives*: Genera un conjunto de alternativas tomando entre aleatorias y prometedoras de una iteración anterior, si es la primera iteración solo toma de las aleatorias.

3. *Machine Learning Model*: El modelo ML sustituto estima los valores donde puede haber mejora esperada en cada alternativa, para luego pasárselo a la función de adquisición.

4. *Acquisition*: Selecciona las mejores X alternativas según la función de adquisición de mejora esperada, tal y como esta descripta en *A Tutorial on Bayesian Optimization of Expensive Cost Functions* [4]. Esta función optimiza la probabilidad condicional del espacio de atributos de las alternativas generadas, para identificar regiones con alternativas más prometedoras. Los detalles se explican en profundidad en la sección 2.4.3 sobre optimización bayesiana.

5. *DEX Model Objetive Function*: Se obtiene la salida del modelo DEX y calcula la función objetivo, para luego entrenar nuevamente el modelo ML sustituto con la nueva alternativa y su valor de función objetivo.

6. *Iterar*: Se actualizan las alternativas óptimas hasta ese momento, de ellas se generan alternativas vecinas (alternativas prometedoras) y se vuelve a 2.

En el trabajo de BAG-DSM [14], los autores afirman que los resultados experimentales generan al menos una alternativa adecuada en menos de un minuto, incluso para los modelos de decisión más grandes. En la mayoría de los casos, el el tiempo de computación fue menor que eso. El descubrimiento de alternativas se distribuyó por igual en todo el tiempo de ejecución, y la calidad de las alternativas también fue adecuado ya que en la mayoría de los casos, las alternativas generadas podrían ser alcanzadas por menos de 5 cambios de atributo. Finalmente, indican que la relación entre la complejidad del modelo de decisión (más grande) y el tiempo de cómputo en los experimentos fue lineal y no exponencial, lo que significa que además es escalable.

2.3. Preliminares

2.3.1. Espacio de búsqueda de contrafácticos

Dada una instancia seleccionada y predicha con un modelo de ML preentrenado, para ser explicada BayCon genera instancias similares que conducen a la predicción deseada por el usuario, es decir, los contrafácticos. Un enfoque ingenuo es generar todas las posibles combinaciones de los atributos-valor o generar iterativamente instancias aleatorias, descartando las que tienen predicción sin cambios. Sin embargo, para conjuntos de datos con una cantidad considerable de atributos, este espacio de búsqueda puede ser abrumadoramente grande, lo que hace que los enfoques ingenuos sean poco prácticos. Este tipo de problemas se llaman de explosión combinatoria, veamos unos ejemplos;

Ejemplo 1. Supongamos un caso de finanzas, siguiendo el ejemplo citado anteriormente en la sección de introducción (1.3). *"Le negaron un préstamo porque su ingreso anual era de $30,000. Si sus ingresos hubieran sido de $45.000, le habrían ofrecido un préstamo"*[53]. Asumamos una base de datos de prestamos, en una versión reducida de ejemplo, la cual dispone los siguientes atributos: nombre, sexo, edad, nivel de estudios alcanzados, estado civil, ingresos, ahorro, y se encuentran discretizados en rangos para facilitar la explicación.

Analizando las características de los valores a tomar para cada atributo de la base de datos, el espacio de posibilidades de una potencial entrada al algoritmo consiste de

Atributo	Valores posibles	Cant Valores
Nombre	Multivariado	No Aplica
Sexo	Masculino; Femenino	2
Edad	< 21; > 21 y < 64; > 65	3
Nivel de estudios	Primario; Secundario; Universitario	3
Estado civil	Soltero, Casado, Divorciado	3
Ingresos	nulo; bajo; medio; alto	4
Ahorro	nulo; bajo; medio; alto	4

Tab. 2.1: Atributos de auto con sus posibles valores.

$2 \cdot 3 \cdot 3 \cdot 3 \cdot 4 \cdot 4 = 864$ posibles combinaciones, las cuales pueden ser evaluadas fácilmente por un algoritmo exhaustivo. Sin embargo, realizar la misma exploración se vuelve imposible con modelos más grandes y complejos.

Ejemplo 2. Tomemos un ejemplo con un modelo más grande [14], más complejo pero realista en el dominio de la agricultura [2] Tabla 2.2. Este conjunto de datos, tiene como objetivo evaluar la productividad primaria de campos agrícolas teniendo en cuenta las propiedades del suelo, los aspectos ambientales, las propiedades de los cultivos y las opciones de gestión.

La completitud o correctitud de esta base de datos se encuentra fuera del alcance de este trabajo. Este ejemplo se incluye aquí para ilustrar dos puntos importantes;

- Primero, después de evaluar un campo agrícola y evaluar su productividad primaria, un agricultor atento haría preguntas como: ¿Qué puedo hacer para mejorar la productividad? ¿Qué puede salir mal y degradar la productividad? Tales preguntas son muy relevantes y deben estar respaldadas por métodos y algoritmos apropiados.

- En segundo lugar, este modelo tiene 26 atributos de entrada y, en consecuencia, un enorme espacio de entrada: $2^4 \cdot 3^{18} \cdot 4^4 \approx 1,59 \cdot 10^{12}$. La búsqueda exhaustiva de soluciones es claramente inviable y se necesitan algoritmos más eficientes.

Una estrategia más apropiada es realizar la búsqueda basada en el registro de contrafactuales previamente generados y evaluados. Estos puntos de datos se pueden utilizar para mapear el espacio de búsqueda y el comportamiento del modelo ML [14]. Establecido en esta aproximación, se pueden generar contrafactuales prometedores de manera más eficiente. Para ello utilizamos optimización bayesiana, que puede ser un vehículo para realizar una búsqueda informada de este tipo de forma estocástica, la cual se explica más en detalle en la sección de metodología (2.4).

Atributo	Descripción
pH	Soil pH (pH-CaCl2)
C/N ratio	C/N ratio
SOM	Soil organic matter (SOM)
P	Major element contents of soil (P)
K	Major element contents of soil (K)
Mg	Additional element contents of soil (Mg)
CEC	Cation exchange capacity (CEC)
Salinity	Salinity
Bulk density	Soil bulk density
Rooting depth	Rooting depth (depth till limitation of root growth)
Clay content	Share of clay in the soil structure
Groundwater Table Depth	Groundwater Table Depth
Precipitation	Annual cumulative precipitation
Temperature	Length of the temperature growing period (degree days)
Altitude	Altitude meters above sea level
Slope degree	Slope degree
Number of crops	Average number of crops in rotation
% legumes	Share of years when legumes have been sawn
% CaC,CoC,GM	Share of years with catch/cash/genetically-modified crops
Stocking rate	Stocking rate (LU/ha/year)
Mineral	Mineral nitrogen fert. (kg N ha-1 y-1)
Organic Nitrogen fert.	Organic nitrogen fertilisation (kg N ha-1)
Chemical	Pest management with chemical control
Physical	Pest management with physical prevention
Biological	Pest management with biological control agents
Irrigation	Irrigation

Tab. 2.2: Atributos de ejemplo para la evaluación de la productividad primaria de los campos de agricultura [2]

2.3.2. Características deseables de los contrafácticos

El pipeline de optimización de BayCon está diseñado para producir explicaciones contrastivas de la más alta calidad, tanto en lo que respecta a sus propiedades técnicas como sociales. Con este fin, nuestro método se adhiere a las últimas pautas que prescriben cómo generar contrafactuales deseables [28];

¿Qué es plausible? BayCon optimiza la plausibilidad minimizando la distancia de los contrafácticos a la instancia explicada, además de extraer automáticamente las restricciones de los atributos del conjunto de datos de entrenamiento subyacente. Además, nuestro método permite al usuario especificar atributos inmutables (como la edad) e indicar valores de los atributos que no son válidos, por ejemplo, número fraccionario de habitaciones en una casa. Todas estas restricciones se utilizan para guiar el muestreo cuasi-aleatorio de características en la sección 2.4.4.3.

Esparcitud. Los contrafactuales deben esforzarse por ajustar el menor número posible de características para hacer explicaciones concretas y cortas, por lo tanto atractivas para los humanos [28]. Sin embargo, el nivel deseado de esparcitud puede depender del usuario

y del conjunto de datos, por lo tanto, incorporamos la cantidad de valores de atributos alterados en la función de optimización utilizada por BayCon. Adicionalmente, el usuario puede especificar el número máximo de atributos alterados.

Factibilidad. Los contrafactuales deben ser factibles y accionables [40]. En particular, los contrafactuales fuera de distribución, que pueden representar el 36 % de todos las explicaciones generadas para algunos métodos – deben ser evitados [30]. BayCon utiliza Local Outlier Factor (LOF) [3] para evitar que tales contrafactuales con valores atípicos se presenten como explicación.

Testeo comparativo. BayCon es comparado contra explicadores contrafactuales de ultima generación en seis conjuntos de datos disponibles públicamente, utilizando métricas de evaluación bien definidas; usamos una combinación de puntajes basados en la distancia de los contrafácticos a la instancia a explicar, como así también medimos el tiempo de ejecución de los métodos comparados para generar los contrafácticos y la cantidad que generaron. Dichos puntajes se encuentran explicados en la siguiente sección.

2.3.3. Función objetivo a optimizar

Para evaluar la calidad de las explicaciones contrafactuales generadas, diseñamos una función objetivo adecuada presentada a continuación:

$$F(\bar{c}, \bar{x}) = S_x * S_y * S_f \tag{2.1}$$

Captura: (1) la distancia en el espacio de características, (2) la distancia en el espacio de salida, y (3) el número de características alteradas, todo escalado al rango [0, 1]. Luego del ejemplo a continuación se explican más en detalle.

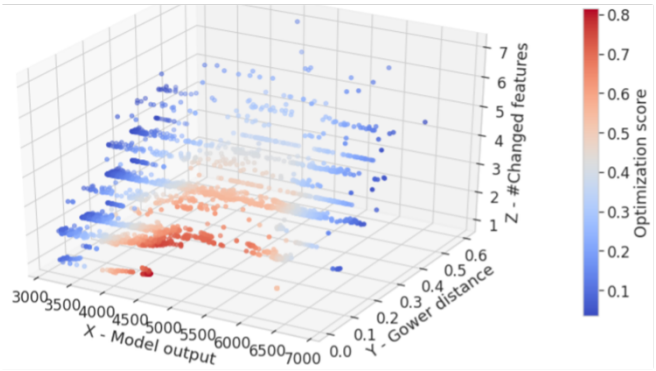


Fig. 2.2: Ejemplo de los puntajes de optimización de BayCon

La Figura 2.2 muestra puntajes de optimización de ejemplo para el conjunto de datos Bike, Tabla 3.1. Cada punto del gráfico es un contrafáctico candidato:

- El eje x representa las predicciones del modelo ML para el que estamos generando contrafactuales;
- El eje y muestra la distancia Gower[1] entre cada contrafactual y la instancia explicada;
- El eje z captura el número de características modificadas;
- El color del marcador indica la puntuación del objetivo de optimización dada por la Ecuación 1 (cuanto más alto, mejor).

En este ejemplo, la instancia explicada se predice como 3141 (bicicletas alquiladas) y el rango de salida deseado (proveído por el usuario) se establece en [4500, 5000]. La figura muestra que:

1. Los puntajes de optimización para los contrafactuales cuyas predicciones (eje y) están fuera del rango especificado por el usuario tienen valor cerca de 0 y aumentan a medida que la salida del modelo se acerca al rango deseado;

[1] Gower es utilizado para espacios de atributos mixtos. Para los atributos categóricos, revisa si las dos características tienen valores idénticos; el componente de distancia es 0 si los atributos son los mismas y 1 en caso contrario. Para atributos numéricos, calcula el valor absoluto de la diferencia entre los valores comparados. Luego se normaliza entre la cantidad de atributos.

2. Las puntajes de optimización disminuyen a medida que aumenta la distancia Gower;
3. Las puntajes de optimización son mayores para los contrafactuales que requieren un menor número de características a cambiar;

A continuación, pasaremos a definir en detalle cada uno de los puntajes que mencionamos previamente;

Similitud en el espacio de atributos (S_x). Calculamos la distancia de Gower entre la instancia explicada $\bar{x}$ y un candidato contrafáctico $\bar{c}$ en nuestra función de optimización (S_x en la Ecuación 3.2), luego lo restamos de 1. Esto efectivamente invierte los valores de Gower; S_x es 1 si las instancias comparadas son idénticas, y va acercándose a 0 a medida que se aleja de ella.

$$S_x(\bar{c}, \bar{x}) = 1 - d_{gower} \tag{2.2}$$

Similitud en el espacio de salida (S_y). Para tareas de clasificación, S_y es 1 si la salida del modelo ML predice el contrafáctico candidato según lo solicitado por el usuario, y 0 en caso contrario. Para problemas de regresión, definimos S_y como:

$$S_y = \begin{cases} 1 & y_c \in [y_{min}, y_{max}] \\ 1 - \frac{|y_c - d|}{|y_x - d| + \theta} & otherwise, \quad where \end{cases} \tag{2.3}$$

$$d = \begin{cases} y_{min} & if \mid y_c - y_{min} \leq \mid y_c - y_{max} \mid \\ y_{max} & otherwise \end{cases} \tag{2.4}$$

- y_x es el resultado del modelo ML para la instancia explicada;
- y_c es la salida del modelo ML para el contrafáctico candidato;
- $[y_{min}, y_{max}]$ es el rango de salida objetivo especificado por el usuario;

Si y_c está en el rango deseado, $S_y = 1$ (el valor máximo). De lo contrario, S_y captura la proximidad de y_c a los bordes (calculado a través de d) del rango deseado. S_y está diseñado para estar dentro del intervalo [0,1].

Proporción de atributos alterados(S_f). Este objetivo simplemente cuenta la cantidad de atributos en el contrafáctico candidato que son diferentes cuando son comparados con los atributos de la instancia a explicar. Este puntaje también se encuentra en el rango [0, 1]

$$S_f(\bar{c}, \bar{x}) = \frac{\#\text{ of different features between } \bar{c} \text{ and } \bar{x}}{\text{Overall \# of features}} \tag{2.5}$$

Para comparar, MOC [9] formaliza la búsqueda de contrafácticos como un problema de optimización multi-objetivo, con generación de contrafácticos mediante Nondominated Sorting Genetic Algorithm II (NSGA-II) [11]. Los objetivos que busca optimizar MOC son;

o_1 : cercanía de la predicción con el objetivo deseado.

o_2 : cercanía con la instancia a explicar en el espacio de atributos

o_3 : cantidad de atributos cambiados

o_4 : plausibilidad de los candidatos contrafácticos basados en la distribución de probabilidad en los valores de los atributos

BayCon replica los objetivos o_1, o_2 y o_3 con los puntajes mencionados previamente: S_y, S_x, y S_f respectivamente. El objetivo o_4 es abordado implícitamente usando el filtro de Local Outlier Filtering (LOF) [3], el cual mide la desviación local de la densidad de una muestra dada con respeto a sus vecinos para descartar los valores atípicos.

2.4. Metodología

En esta sección presentamos la implementación y los detalles de BayCon para generar contrafácticos.

2.4.1. Arquitectura

La arquitectura de BayCon se ha basado fuertemente en la del estudio BAG-DSM [14], dicha arquitectura está presentada en detalle en los trabajos relacionados.

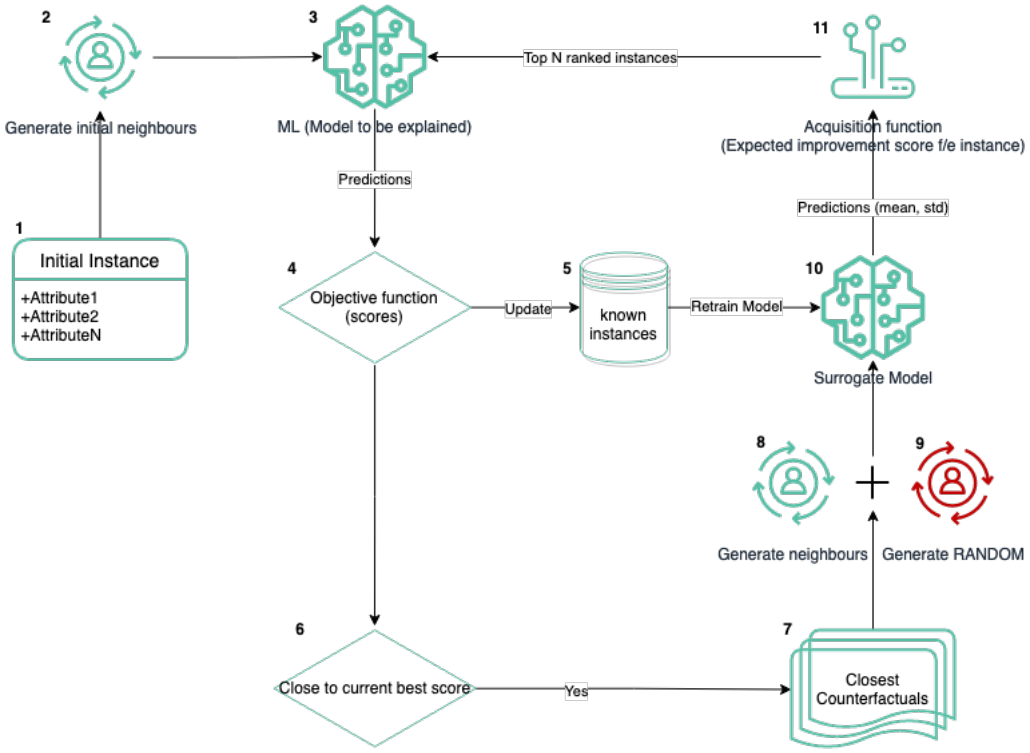


Fig. 2.3: Detalle gráfico de la arquitectura de BayCon

Presentamos cada paso de la arquitectura de BayCon. Algoritmo 1 captura la implementación de BayCon en más detalle. Las estrategias de generación de instancias, tanto la inicial, la de vecinos, como la aleatoria, se encuentran descriptas a lo largo de la generación de contrafácticos en la sección 2.4.4.

1. *Dada una instancia inicial $\bar{x}$ con su predicción del modelo ML a explicar*

2. *Generación inicial de vecindario*: Sección 2.4.4.1

3. *Modelo de IA a ser explicado*: Junto a la instancia inicial y su predicción de este modelo, estamos buscando generar instancias que cambien su predicción, es decir, los contrafácticos.

4. *Función objetivo*: Sección 2.3.3

5. *Actualizar las instancias conocidas*: Ahora se conoce la predicción de las instancias por el modelo ML a explicar, por lo tanto se actualizan y se reentrena el modelo ML sustituto con las instancias y sus valores de la función objetiva.

6. *Filtrado*: Sección 2.4.4.6

7. *Actualizar instancias óptimas*

8. *Generación de vecinos*: Sección 2.4.4.1

9. *Generación aleatoria*: Sección 2.4.4.3

10. *Modelo sustituto*: Sección 2.4.3.1 Se basa en la optimización bayesiana, Sección 2.4.3

11. *Función de adquisición*: Sección 2.4.3.2

12. *Iterar*: Se itera hasta que el puntaje de los contrafacticos obtenido no tenga una mejora. Máximo 100 iteraciones.

2.4.2. Algoritmo

Algorithm 1 Detalles de implementación de BayCon.

Input: black-box-model **f**, instance to be explained **x***, desired prediction **p**, training data X_T.
Output: Counterfactuals ***CFs***

$X = generate\_neighbourhood(x^*)$
$y = f(X)$ ▷ predict neighbourhood
$S_x = objective\_function(X, y, p)$ ▷ calculate scores
$X_k, y_k = update\_known\_instances(X, y)$
$g = RandomForest(X_K, S_X)$ ▷ train surrogate model

while *continue_search* **do**
    $CF = select\_counterfactuals(X_K, y_K)$
    $CF_b, S_b = select\_best(CF, S_X)$
    $X = generate\_neighbourhood(CF_b)$
    $X+ = random\_generation(S_b)$
    $\mu, \sigma = g(X)$
    $X_R = acquisition\_function\_rank(X, \mu, \sigma)$
    $y = f(X_r)$ ▷ Get black-box predictions
    $S_x+ = objective\_function(X_R, y, p)$
    $X_k = update\_known\_instances(X_R, y)$
    $g.retrain(X_k, S_x)$ ▷ update surrogate model
end while

$y = f(X_p)$ ▷ get black-box predictions
$CFs+ = update\_with\_counterfactuals\_from(X_p, y)$
$CFs = LOF\_filter(CFs, X_T)$

2.4.3. Optimización Bayesiana

Utilizamos optimización bayesiana [4], la cual permite utilizar creencias previas sobre un problema para ayudar a navegar el muestreo. Según los autores, la optimización bayesiana emplea la técnica bayesiana de establecer una prioridad sobre la función objetivo y combinarla con evidencia para obtener una función posterior. Esto permite una selección basada en la utilidad de la próxima observación a realizar sobre la función objetivo, que debe tener en cuenta tanto la exploración (muestreo de áreas de alta incertidumbre) como la explotación (muestreo de áreas que probablemente ofrezcan mejoras sobre la mejor observación actual).

Repasemos el teorema de Bayes, el cual describe una forma de calcular la probabilidad condicional de un evento:

$$P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}. \tag{2.6}$$

Esta ecuación se puede simplificar eliminando el valor de normalización de $P(B)$ y describiendo la condicional probabilidad como una cantidad proporcional.

$$P(A \mid B) \propto P(B \mid A)P(A). \tag{2.7}$$

Donde $P(A|B)$ es la probabilidad posterior, $P(B|A)$ es la posibilidad (también llamado likelihood) y $P(A)$ la probabilidad anterior (por lo tanto, *posterior* $\propto$ *likelihood* $*$ *anterior*). Siguiendo esta idea, podemos generar alternativas específicas y evaluarlas usando nuestra función objetivo F, Ecuación 2.1. Al generar iterativamente tales alternativas, se pueden crear los datos de aprendizaje D; en nuestro caso, D consta de n contrafactuales observados y sus predicciones del modelo a explicar:

$$D = \{(\bar{c_1}, F(\bar{c_1}, \bar{x}), (\bar{c_2}, F(\bar{c_2}, \bar{x})), \ldots, (\bar{c_n}, F(\bar{c_n}, \bar{x}))\}. \tag{2.8}$$

Estos datos definen la probabilidad anterior para el problema específico. La función de probabilidad se define como la probabilidad de observar los datos dada la función objetivo F; $P(D \mid F)$. Esta función de probabilidad cambiará a medida que se generen más alternativas:

Finalmente, para nuestro problema la optimización bayesiana es la probabilidad posterior de una función F dados datos D (o evidencia) es proporcional a la probabilidad de D (dado F) y la probabilidad anterior de F:

$$P(F \mid D) \propto P(D \mid F)P(F). \tag{2.9}$$

Donde $P(F \mid D)$ representa las expectativas actualizadas sobre la función objetivo desconocida [4]. Este paso también puede interpretarse como la estimación de la función objetivo con una función sustituta.

2.4.3.1. Modelo sustituto

Para estimar el posterior de nuestra función objetivo (Ecuación 4.4), empleamos un modelo sustituto. Este es un modelo ML típicamente aprendido con algoritmos de regresión basados en un Proceso Gaussiano (GP) porque dichos modelos brindan acceso a la totalidad distribución de probabilidad [47][55]. Explotando la media y la desviación

estándar de la distribución de la salida, se puede equilibrar la explotación (media más alta) y exploración (desviación estándar más alta).

Dado que los GP son computacionalmente costosos: complejidad $O(n^3)$ - Se pueden usar en su lugar los ensembles de modelos de regresión como Random Forest (RF) [25]. En cuyo caso, la media y la varianza se calculan en base a las predicciones de todos los modelos individuales dentro del conjunto.

La entrada del modelo sustituto se encuentra definido de la siguiente manera:

$$input = [\Delta k_1, \ldots, \Delta k_n, count(\Delta k), d_{gower}] \tag{2.10}$$

- Δk_i representa la distancia entre $\bar{c}$ y $\bar{x}$ en el atributo i,
- $count(\Delta k)$ es la cantidad de atributos cambiados en $\bar{c}$ comparados con $\bar{x}$,
- d_{gower} es la distancia de Gower entre $\bar{c}$ y $\bar{x}$

Por lo tanto, para cualquier entrada dada, el modelo sustituto predice una estimación para nuestro puntaje de optimización. En un principio este input constaba solo de $[\Delta k_1, \ldots, \Delta k_n]$, pero dado que estos valores podrían ser relativamente bajos en los atributos del potencial contrafáctico, y sin embargo presentar muchos atributos cambiados en total, eso nos afectaría el puntaje S_f, por lo tanto le agregamos $count(\Delta k)$ para que el modelo ML sustituto aprenda que a menor cantidad de atributos cambiados, corresponden contrafácticos mas prometedores. Finalmente, buscamos maximizar el puntaje S_x, es decir, minimizar la distancia de Gower del potencial contrafáctico con la instancia a explicar, para reflejarlo se le agrego d_{gower} como último atributo.

2.4.3.2. Función de adquisición

La media $\mu(\bar{S_c})$ y la varianza $\sigma(\bar{S_c})$ calculadas en la salida del modelo sustituto se utilizan como entrada para la función de adquisición, que es responsable de seleccionar los contrafactuales más prometedores. Esta función optimiza la probabilidad condicional del espacio de características para identificar regiones con contrafactuales más prometedores. BayCon usa la *Mejora esperada* como su función de adquisición [36], observar la Ecuación 4.6. En nuestros experimentos, la constante que controla el equilibrio entre la búsqueda global y la optimización local (es decir, exploración/explotación) se establece como $\xi = 0{,}01$ [32].

Intuitivamente, esta función de adquisición comprueba la mejora que aporta cada candidato contrafáctico con respecto al máximo valor conocido S_b (i.e., $\mu(S_c) - S_b$), y escala esta mejora con respecto a la incertidumbre dada por $\sigma(S_c)$.

La función de adquisición de mejora esperada se define como [36]:

$$EI(\bar{S_c}, S_b) = \begin{cases} \mu((\bar{S_c}) - S_b - \xi)\Phi(Z) + \sigma(\bar{S_c})\Phi(Z), & if \quad \sigma(SM(\bar{x})) > 0 \\ 0, & if \quad \sigma(\bar{S_c}) = 0 \end{cases}$$
$$Z = \begin{cases} \mu(\bar{S_c} - S_b - \xi)/\sigma(\bar{S_c}) > 0 \\ 0, \qquad\qquad if \quad \sigma(\bar{S_c}) = 0 \end{cases} \tag{2.11}$$

$\bar{S}_c$ es el resultado del modelo sustituto para el contrafactual $\bar{c}$; S_b es el máximo valor conocido de contrafactuales candidatos vistos hasta ahora; $\Phi(\cdot)$ es la función de distribución acumulativa normal; y ξ es una constante que controla el equilibrio entre la búsqueda global y la optimización local (es decir, exploración/explotación).

Si dos contrafácticos tienen una media similar, el de mayor incertidumbre es el preferido por la función de adquisición. Esto se debe a que mayor incertidumbre corresponde un mayor espacio de exploración, lo cual incrementa la posibilidad de mejorar el posterior de nuestra función objetivo (Ecuación 2.1). Una representación visual de este proceso en un ejemplo de un atributo se presenta a continuación:.

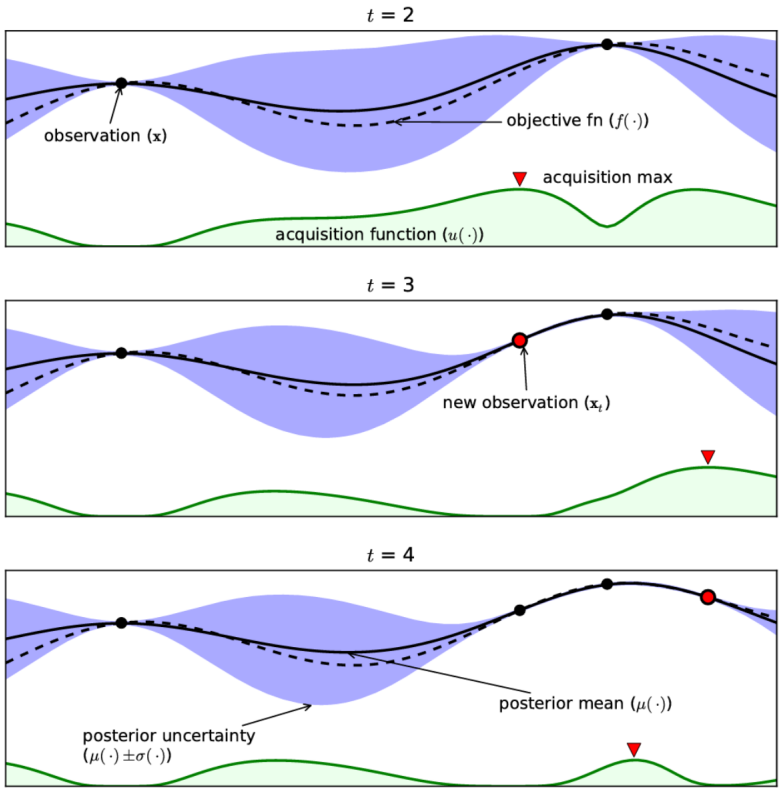


Fig. 2.4: Ejemplo esquemático de optimización bayesiana [32]

Figura 2.4 presenta un unidimensional — un solo atributo dado a lo largo del eje x. La gráfica muestra una aproximación de la función objetivo usando un modelo sustituto sobre cuatro iteraciones de muestreo de la función objetivo. La figura también muestra la función de adquisición, que es capturada por el gráfico verde. Esta función tiene un valor alto donde el modelo sustituto predice un objetivo de alto valor (explotación) o donde el la incertidumbre de predicción es alta (exploración), por lo tanto, las regiones de alto valor objetivo y alta incertidumbre se exploran primero. Cabe destacar que el área del extremo izquierdo de la gráfica permanece sin muestrear; si bien tiene una alta incertidumbre, también se predice (correctamente) para ofrecer poca mejora sobre la observación de mayor valor.

2.4.4. Generando contrafácticos

En conjunto con la optimización bayesiana, estrategias para la generación de contrafácticos es una de las partes cruciales de nuestra arquitectura, en el estudio de MOC [9] utilizaron NGSA-II [11], nosotros utilizamos otro enfoque basado en sampleo alrededor de la vecindad a la instancia original a explicar.

Supongamos que buscamos generar contrafácticos $\bar{c}$ a una instancia a explicar $\bar{x}$ que consta de solo un atributo-valor x_1. Ya que buenos contrafácticos se encuentran cerca de sus instancias a explicar en el espacio de atributos [28], nos alcanza con muestrear y filtrar cerca del valor x_1. Una estrategia es filtrar dentro de un radio r alrededor de $\bar{x}$, es decir $c_1 \in (x_1 - r_1; x_1 + r_1)$, actualizar r_1 a medida que se encuentran soluciones más cercanas, achicando r en cada descubrimiento y así acercándose estocásticamente a $\bar{x}$.

Expandiendo la instancia a explicar $\bar{x}$ a n atributos, para los contrafácticos $\bar{c}$ necesitamos filtrar varios rangos de atributos c_i. Al radio r corresponde la distancia normalizada Gower entre $\bar{c}$ y $\bar{x}$, esto sería similar a un circulo/esfera/hiperesfera centrada en $(x_1, ..., x_n)$. A medida que la cantidad de atributos aumenta, corresponde un aumento de dimensiones de búsqueda y un espacio más grande R^N. Pero solo debemos buscar dentro de una hiperesfera con radio $r = d_{Gower}(\bar{x}, \bar{c})$, es decir, $\bar{c} \in (x_1 \pm r_1, ..., x_n \pm r_n) \subset R^N$. Por lo resaltado anteriormente [28], es en dicha hiperesfera es donde se encontrarán mejores contrafácticos. Nuestra estrategia se basa en disminuir dicho radio a medida que se encuentran contráfacticos $\bar{c}$ más cercanos a la instancia a explicar $\bar{x}$, reduciendo la hiperesfera estocásticamente y dejando muestras cada vez más cercanas.

2.4.4.1. Generación inicial de vecindario

Dada la suposición que los mejores contrafácticos deben estar cerca de la instancia a explicar, nuestra búsqueda inicial se centra en su vecindad. Para generar este espacio, muestreamos cada atributo con valores al azar, utilizando una distribución normal. Dicha normal se encuentra centrada alrededor de la instancia inicial. Para los atributos númericos se la utiliza truncada, es decir limitada en los rangos de cada atributo, ya que el valor original de la instancia a explicar podría tener atributos con valores muy cerca de los limites de los rangos. Los atributos categóricos se muestrean uniformemente a lo largo del conjunto de valores posibles.

2.4.4.2. Explorando mejores vecindarios

Ya que buenos contrafácticos deben provenir de regiones densas, y no de outliers, exploramos vecindarios de explicaciones con mejores puntajes. Reutilizamos el mismo procedimiento de generación inicial de vecindario (aplicado a la instancia inicial), pero con una normal centrada en los valores-atributos de los mejores contrafácticos encontrados hasta el momento.

2.4.4.3. Sampleo aleatorio de atributos

Para permitir un mayor grado de exploración y no quedar sesgados a los contrafácticos ya encontrados, generamos contrafácticos aleatoriamente, para ello muestreamos valores de los atributos numéricos uniformemente al azar dentro de sus rangos. Los atributos categóricos se muestrean uniformemente sobre el conjunto de valores posibles.

2.4.4.4. Redondeo

Para evitar contrafácticos indistinguibles que solo difieren menos de un n-ésimo lugar decimal para los atributos numéricos, realizamos la discretización de k-bins con bins de igual ancho. Usamos k = 100 para nuestros experimentos, lo que proporciona la diferencia mínima de 1 % relativa al rango de cada atributo. Esto fue implementado debido a que muchos valores sampleados de las distribuciones, al ser de punto flotante, quedaban cercanos entre ellos y los detectaba como contrafácticos diferentes, cuando en realidad, son casi indistinguibles y a términos prácticos, son el mismo.

2.4.4.5. Selección de atributos a ser modificados

Para aumentar la esparcitud, es decir, cambiar la menor cantidad posible de características por contrafáctico, seleccionamos aleatoriamente atributos para actualizar en función de una distribución sesgada, donde la probabilidad de cambiar n características es el doble que la de cambiar n+1. Sólo las características seleccionadas son luego actualizadas utilizando los procedimientos descritos en los anteriores pasos (generación de vecindad o muestreo aleatorio).

2.4.4.6. Filtro

BayCon es un algoritmo iterativo; En cada paso, removemos los candidatos a contrafáctico cuya puntuación está por debajo del mejor hasta el momento. Además, antes de generar las explicaciones con contrafácticos, eliminamos aquellos que se encuentran fuera de distribución con LOF, el cual mide la desviación local de densidad de cada contrafáctico con respecto a su entorno de conjunto de entrenamiento. Las explicaciones que tienen una densidad sustancialmente menor que sus vecinos por lo tanto, se eliminan. Para este proposito, usamos la implementación de scikit-learn con sus parametros por defecto. [3]

2.4.5. Detalles de implementación

El desafío inicial fue revisar y entender el código del trabajo de BAG-DSM [14] y asociarlo con su arquitectura, la cual esta escrita en *Python/Java* y hacerlo funcionar. Luego, pasó a un proceso de refactorizado del código, incorporando al mismo tiempo estructura y un modelo orientado a objetos, con el objetivo de remover su acoplamiento con el modelo DEX, cambiarlo por un modelo de IA, y aumentar la cohesión de las responsabilidades de cada parte. Realizando este trabajo, extrajimos varios conceptos/objetos. Los resumimos a continuación:

- *Target(target_type, target_feature, target_value):* Representa las diferentes clases de objetivos que se pueden elegir; clasificación o regresión. También permite especificar cual es el atributo sobre el cual queremos realizar predicciones con el modelo objetivo, y su valor deseado.
- *DataAnalyzer(X, Y, feature_names, target, categorical_features):* Realiza el análisis de los datos de entrenamiento (X e Y) del modelo a explicar, junto con un listado de los nombres de los atributos y el del atributo objetivo (target). Este calcula los rangos de valores en los cuales se mueven los atributos de los datos de entrenamiento, entre otras funcionalidades. Además permite marcar manualmente los atributos categóricos/numéricos.

- *InstancesGenerator(initial_instance, data_analyzer, score_calculator):* Encargado de la generación de los contrafácticos, para ello realiza diferentes estrategias de generación de vecinos a la instancia que se busca explicar, se ayuda del analisis de datos para poder samplear las distribuciones de los datos de entrenamiento.

- *ScoreCalculator(initial_instance, initial_prediction, target, data_analyzer):* Realiza el calculo de los puntajes de S_x, S_y, S_f y F (Ecuación 2.1) respecto de las instancias generadas y la instancia inicial con su predicción a explicar. Para calcular S_y en el caso de que el objetivo sea de regresión, utiliza el rango del atributo objetivo del análisis de los datos de entrenamiento.

- *InstancesInfo(instances, score_calculator, model):* Almacena la información de las instancias sampleadas con *InstancesGenerator*. Para cada una de las instancias realiza la predicción con el modelo objetivo a explicar, y luego calcula y almacena el puntaje de cada una con *ScoreCalculator*.

- *SurrogateRanker(surrogate_model, initial_instance, target):* Guía la búsqueda de contrafácticos. De las instancias generadas y que aun se desconoce su predicción del modelo ML a explicar, toma las primeras n mejores instancias que la función de adquisición prioriza donde puede haber una mejora. Para ello utiliza la predicción del modelo ML sustituto sobre la mejora esperada en cada instancia. Realiza el proceso de entrenamiento del modelo ML sustituto cada vez que se van conociendo las predicciones del modelo ML a explicar, sobre las instancias generadas.

- *AcquisitionFunction:* Función de adquisición de mejora esperada, tal y como esta descripta en *A Tutorial on Bayesian Optimization of Expensive Cost Functions* [4].

A estos objetos se los utiliza para realizar una búsqueda iterativa hasta no encontrar instancias con mejores puntajes, finalmente se filtran las que se encuentran fuera de distribución de los datos utilizando LOF [3] y se devuelven las instancias contrafácticas encontradas, es decir, las explicaciones contrafácticas de la instancia inicial a explicar. Se puede ver el detalle de implementación del proceso iterativo en el Algoritmo 1.

Para gestionar todo el código con la implementación, como así el historial, código inicial, y la incorporación de los diferentes conceptos, se utilizo la herramienta de versionado *git* desde el comienzo. Todo el proyecto se encuentra disponible públicamente a través de Github: github.com/piotromashov/baycon

Uno de los desafíos que cabe destacar a nivel de implementación, es el de haber incorporado la biblioteca $NumPy$ y repensar todas las operaciones para que sean matriciales, explotando su potencial de correr en varios hilos de ejecución, logrando así una mejora en el tiempo de calculo de varias instancias al mismo tiempo.

3. EXPERIMENTOS

En esta sección presentamos los diversos experimentos que realizamos con el código que mostramos en la metodología, detallamos su configuración, indicamos las características de los diferentes conjuntos de datos elegidos y realizamos un análisis de los resultados obtenidos.

3.1. Hardware & Software

Realizamos comparaciones de BayCon con otros métodos de generación de contrafácticos, en seis conjuntos de datos de la vida real. Nuestro método está implementado en Python 3.6 y se basa en gran medida en scikit-learn [39]. Todos los experimentos se realizaron en un procesador Intel Core i9 de 3,70 GHz. con 128 GB de RAM. Le impusimos un tiempo límite de 15 minutos de ejecución para cada experimento.

La implementación de BayCon, el código de experimentación, incluidos los conjuntos de datos procesados, y el análisis de los resultados, están a disposición públicamente en GitHub: github.com/piotromashov/baycon.

3.2. Conjuntos de Datos

Para realizar los experimentos utilizamos bases de datos publicas, utilizamos 3 de regresión, 3 de clasificación, con un grado creciente de cantidad de atributos, para poder comparar la escalabilidad a problemas con espacios de búsqueda más complejos.

Dataset	Atributos(Num/Cat)	Tipo	Muestras
Diabetes	8/0	Cls	768
kc2	22/0	Cls	522
Biodeg	41/0	Cls	1055
Bike	7/3	Reg	730
House Sales	19/2	Reg	21613
Tecator	125/0	Reg	240

Tab. 3.1: Conjuntos de datos utilizados para la evaluación experimental.

Todos los conjuntos de datos están disponibles en línea; el conjunto de datos de Bike se puede descargar del repositorio de la UCI3 y los demás están disponibles a través de el repositorio OpenML [51].:

- *Diabetes, Pima Indians Diabetes Database*
  National Institute of Diabetes and Digestive and Kidney Diseases.
  Fuente: openml.org/d/37
- *Kc2, Predicción de defectos en software*
  Dataset de datos métricos de defectos en programas de la NASA.
  Fuente: openml.org/d/1063

- *Biodeg, Actividad de Relaciones de Estructura Cuantitativa*
  Modelos de relación de actividad para biodegradabilidad de productos químicos.
  Fuente: openml.org/d/1494

- *Bike, Dataset de bicicletas compartidas*
  Recuento de alquiler de bicicletas con información meteorológica y estacional.
  Fuente: archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

- *House Sales, Venta de casas*
  Precios de venta de casas para el condado de King, incluyendo Seattle.
  Fuente: openml.org/d/42731

- *Tecator, Analizador de alimentos Tecator Infratec*
  Predice el contenido de grasa de una muestra de carne sobre la base de su absorción de espectro infrarrojo cercano.
  Fuente: openml.org/d/505

3.3. Configuración experimental

Comparamos nuestro método propuesto con una búsqueda de contrafácticos exhaustiva de fuerza bruta implementada en FAT Forensics [48] , y MOC basado en su implementación oficial [9]. FAT Forensics solo brindó explicaciones para el conjunto de datos de Diabetes dado el límite de tiempo impuesto de 15 minutos, por lo tanto no aparece en nuestra comparación. MOC, por otro lado, generó explicaciones para todos los conjuntos de datos excepto para House (probablemente debido al tamaño de su conjunto de entrenamiento) ya que es un método de última generación.

Para la comparación utilizamos 6 diferentes conjuntos de datos, tres de clasificación (Cls), y 3 de regresión (Reg), más detalles en Tabla 3.1.

Para cada conjunto de datos de clasificación, seleccionamos 10 instancias aleatorias para ser explicadas, generando sus explicaciones con contrafactuales 3 veces para dar cuenta de la aleatoriedad (es decir, 30 ejecuciones por conjunto de datos).

Para cada conjunto de datos de regresión, seleccionamos 3 instancias iniciales, una para cada percentil de la variable de salida: la mediana así como los percentiles 25 y 75 (y_x en Ecuación 3.1). A continuación, generamos explicaciones para 4 rangos objetivo deseados: aumentar, disminuir, estar en un intervalo por encima de $(y_x + a, y_x + b)$ y estar en un intervalo por debajo de $(y_x - b, y_x - a)$ la predicción de la instancia explicada, con a y b definido en la Ecuación 3.1. Cada experimento se repitió 3 veces (es decir, 36 ejecuciones por conjunto de datos).

$$a = 0{,}5 * y_x; b = 0{,}75 * y_x \tag{3.1}$$

Explicamos las predicciones de dos modelos de caja negra: Random Forest (RF) y Support Vector Machine (SVM). Los modelos fueron entrenados con todos los datos excluyendo las instancias a explicar. Dado que las SVM pueden ser sensibles a la escala de los atributos y la parametrización del modelo, aplicamos la normalización mínima-máxima a los atributos de entrada y ajustamos los parámetros del modelo mediante una validación cruzada triple (3-fold cross validation) en los datos de entrenamiento.

3.4. Resultados

Hemos tenido resultados sobresalientes que están relacionados con el escalado a conjuntos de datos más grandes. Por ejemplo, para el conjunto de datos más grande - 125 atributos (Tabla 3.2), logramos tener un primer (mejor) contrafáctico en 0.1 segundos y generar otros 43k en promedio en 1:30 minutos. En experimentos similares, los métodos de trabajos relacionados tenían problemas de escalado, es decir, MOC y FAT Forensics no no terminaron en la ventana dada de 15 minutos de ejecución.

En términos de "tiempo total de cómputo" y "tiempo para el primer (mejor) contrafáctico", nuestros experimentos mostraron que BayCon es más rápido que MOC (al menos 10 veces, Tabla 3.2). También fue más rápido que FAT Forensics, un generador contrafactual exhaustivo de fuerza bruta - más lento que MOC y BayCon. Sin embargo, un trabajo futuro es el del análisis detallado de la complejidad y la escalabilidad.

	Método	Tiempo Total	Tiempo 1^{er} CF	#Contrafácticos
Diabetes	BC	3.5(1)	0.1(0)	398(173)
	MOC	80.7(40)	1.6(1)	58(28)
kc2	BC	9.0(5)	0.4(1)	2529(1437)
	MOC	192.1(133)	8.5(21)	45(19)
Biodeg	BC	13.2(9)	0.3(0)	1138(755)
	MOC	302.7(167)	4.5(8)	100(49)
Bike	BC	4.7(2)	0.0(0)	1446(493)
	MOC	47.6(27)	1.1(2)	78(56)
House Sales	BC	26.2(4)	0.2(0)	1723(674)
	MOC	/	/	/
Tecator	BC	83.3(34)	0.1(1)	42949(20154)
	MOC	429.1(98)	155.5(52)	3(1)
Promedio	BC	23.3(9.2)	.18(.33)	8363(3947)
	MOC	210.4(93)	3.16(17)	47(25)

Tab. 3.2: Los valores representan a - $\mu(\sigma)$ - calculado para cada experimento.

La Tabla 3.2 compara BayCon y MOC con respecto al tiempo de cómputo total, el tiempo hasta la primera solución y el número de contrafactuales generados (todos los tiempos se dan en segundos). El tiempo hasta la primera solución para MOC se calculó como el tiempo total dividido por el número de explicaciones generadas. Si bien tal estrategia le da una ventaja a MOC, BayCon la superó en todos los ámbitos. Además, el conjunto de datos House hizo que se agotara el tiempo de corrida para MOC, lo que probablemente se deba al tamaño del conjunto de datos de entrenamiento.

La Tabla 3.3 describe la comparación experimental entre BayCon y MOC utilizando los tres puntajes de evaluación, S_y, S_f, S_x, propuestos en los preliminares en la sección 2.3.3. Se presenta la media y la desviación estándar para cada puntaje y el resultado adjunto de la prueba de rango U de Mann-Whitney, el cual es un test no paramétrico de la hipótesis nula de que la distribución subyacente de la muestra x es la misma que la distribución subyacente de la muestra y, a menudo se utiliza como una prueba de diferencia de ubicación entre distribuciones, en este caso nuestras distribuciones están dadas por los

valores de los experimentos para las muestras de BayCon y MOC. También se indica el tamaño de la muestra, que depende de la cantidad de contrafácticos generados en cada ejecución experimental: un tamaño de muestra de 833 indica que para este experimento comparamos las puntuaciones de 833 explicaciones generadas por MOC con la misma cantidad de explicaciones generadas por BayCon.

Dado que los métodos podían generar un número diferente de explicaciones, solo tomamos los n principales contrafácticos (clasificados por el puntaje de evaluación) con n determinado por el menor número de explicaciones por método generado para una configuración experimental dada. Los valores de p en negrita resaltan los experimentos en los que BayCon superó a MOC con significancia estadística ($p < 0{,}05$). Esto sucedió en la mayoría de los experimentos, a excepción de los conjuntos de datos de Bike y Tecator predichos con un SVM, pero solo cuando se mide por la puntuación S_y. En estos experimentos específicos, MOC encontró mejores contrafácticos en el espacio de salida (S_y), pero BayCon encontró mejores contrafácticos en el espacio de atributos (S_f y S_x). En todos los experimentos, BayCon ofreció contrafácticos con un número menor de atributos modificados y una distancia Gower más pequeña, y aún logrando la predicción objetivo, es decir, la instancia a explicar con una predicción diferente solicitada por el usuario.

Por otro lado, BayCon se diseñó inicialmente para usar Procesos Gaussianos (GP) como modelo sustituto en la guía de búsqueda de contrafácticos. Sin embargo, esa implementación fue más lenta debido a la complejidad computacional de GP (por ejemplo, 20x más lento que Random Forest). RF también estaba funcionando de manera similar a GP, i.e., no hubo diferencias significativas entre los dos implementaciones en las métricas de evaluación. Hallazgos similares se presentaron en el trabajo relacionado para usar RF como un sustituto modelo en problemas de optimización bayesiana, por lo que decidimos continuar con RF. Sin embargo, realizar un análisis exhaustivo de GP vs RF es un trabajo futuro interesante.

Por último, es cierto que las métricas de evaluación propuestas en los preliminares (Ecuación 2.1) son las mismas que utilizamos para la optimización de BayCon. Sin embargo, vale destacar que MOC también usa las mismas métricas con una implementación ligeramente diferente. Además, todas las métricas se basan en el trabajo relacionado sobre generación de contrafácticos [28], lo que significa que las métricas no fueron diseñadas específicamente para poner a MOC en desventaja. Más específicamente, las tres métricas están relacionadas con:

- *Salida del modelo*: por definición, los contrafácticos deben cambiar la salida del modelo hacia un valor deseado. Este elemento está optimizado tanto por MOC como por BayCon. Más específicamente, en MOC se nombra como o_1 (objetivo 1).
- *Distancia de Gower*: una métrica de distancia que funciona tanto con características numéricas como categóricas. Esta misma distancia es utilizada y optimizada tanto por MOC como por BayCon. Más específicamente, en MOC se nombra como o_2 (objetivo 2).
- *Número de funciones modificada*s: cuenta el número de características cambiadas. Esta misma métrica se utiliza y optimiza tanto por MOC como por BayCon. Más específicamente, en MOC se nombra como o_3 (objetivo 3).

Bike												
	RF (muestras = 833)						SVM (muestras = 767)					
	S_y		S_f		S_x		S_y		S_f		S_x	
	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC
μ	1.0	1.0	.90	.69	.98	.69	.72	1.0	.85	.69	.93	.65
σ	0.0	0.0	.01	.07	.03	.49	.23	0	.07	.06	.08	.57
	$p >$,05		**p < 0,001**		**p < 0,001**		$p <$ 0,001		**p < 0,001**		**p <** ,**001**	
Kc2												
	RF (muestras = 528)						SVM (muestras = 512)					
	S_y		S_f		S_x		S_y		S_f		S_x	
	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC
μ	1.0	1.0	.90	.87	.97	.96	1.0	1.0	.87	.67	.99	.81
σ	0.0	0.0	.06	.09	.06	.09	0.0	0.0	.14	.33	.02	.33
	$p >$,05		**p =** ,**046**		**p < 0,001**		$p >$,05		**p < 0,001**		**p < 0,001**	
Diabetes												
	RF (muestras = 1002)						SVM (muestras = 1137)					
	S_y		S_f		S_x		S_y		S_f		S_x	
	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC
μ	1.0	1.0	.84	.70	.94	.89	1.0	1.0	.83	.71	.97	.90
σ	0.0	0.0	.06	.18	.03	.10	0.0	0.0	.08	.14	.02	.08
	$p >$ 0,05		**p < 0,001**		**p < 0,001**		$p >$ 0,05		**p < 0,001**		**p < 0,001**	
Tecator												
	RF (muestras = 6)						SVM (smuestras = 20)					
	S_y		S_f		S_x		S_y		S_f		S_x	
	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC
μ	.95	1.0	.99	.94	.99	.93	.67	1	.99	.85	.99	.92
σ	.05	0.0	.01	.02	.11	.01	.02	0.0	.01	.03	.01	.02
	*Pocas muestras para un test estadístico						$p <$ 0,001		**p < 0,001**		**p < 0,001**	
Biodeg												
	RF (muestras = 1437)						SVM (muestras = 1857)					
	S_y		S_f		S_x		S_y		S_f		S_x	
	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC
μ	1.0	1.0	.95	.94	1.0	1.0	1.0	1.0	.98	.95	1.0	.99
σ	0.0	0.0	.02	.03	.004	.007	1.0	1.0	.01	.03 .01	.01	
	$p >$ 0,05		**p < 0,001**		**p < 0,001**		$p >$ 0,05		**p < 0,001**		**p < 0,001**	
House Sales												
	RF (muestras = 31036)						SVM (muestras = 45797)					
	S_y		S_f		S_x		S_y		S_f		S_x	
	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC	BC	MOC
μ	.46	-	.96	-	.86	-	.47	-	.98	-	.88	-
σ	.28	-	.04	-	.09	-	.26	-	.02	-	.08	-
	*MOC no terminó en el tiempo límite impuesto de 15 minutos											

Tab. 3.3: Resultados de los experimentos con las diferentes bases de datos

4. CONCLUSIONES

Generar contrafácticos para descubrir hipotéticos escenarios predictivos es el estándar de facto para explicar los modelos de aprendizaje automático (IA) y sus predicciones [28]. En este trabajo presentamos BayCon, un generador de contrafácticos basado en sampleo probabilístico de atributos y optimización bayesiana. El objetivo principal de BayCon fue de generar alternativas que requieran el menor cambio en la instancia a explicar para obtener una predicción deseada. BayCon es agnóstico del modelo a explicar, compatible con tareas de regresión y clasificación, y capaz de procesar características numéricas y categóricas, lo cual lo hace el método preferencial dentro de sus trabajos relacionados por su flexibilidad.

Nuestros resultados experimentales demostraron que BayCon es adecuado para la tarea, (e.g., genera una alternativa apropiada en menos de un minuto), para diversos tipos de conjuntos de datos y con complejidad incremental, lo cual prueba su escalabilidad. En comparación con los métodos más avanzados y novedosos, es más eficiente en el tiempo y genera conjuntos de contrafácticos más grandes y diversos (ver la Tabla 3.2). Además, las explicaciones generadas por nuestro algoritmo son de mejor calidad: se colocan más cerca de la instancia a explicar, y requieren menos ajustes de atributos, lo que las hace más similares.

Este tipo de resultados se pueden usar para diversos dominios donde se dispone de una instancia original y se quiere buscar por donde realizar cambios mínimos en los atributos, cumpliendo o acercándose al objetivo propuesto. Esto puede ser muy útil en ámbitos como por ejemplo la agricultura, donde se dispone de (potencialmente) centenares de atributos, y que pueden tomar diversos tipos de valores cada uno [2]. En el contexto de seguridad informática, un tipo de usos posible ser el de *adversarial attacks*, donde los ataques intentan descubrir el mínimo cambio que se deben aplicar a los datos de entrada para causar una clasificación diferente. Esto ha sucedido con respecto a la visión por computadora en sistemas de vehículos autónomos; un cambio mínimo en una señal de alto, imperceptible para el ojo humano, llevó a los vehículos a detectarla como una señal de 45 mph [16]

Una suposición en todos los experimentos fue que todos los atributos pueden ser cambiados. En la realidad esto no siempre es verdad. Por ejemplo, si tuviéramos un modelo de decisión para evaluar la calidad del suelo, el tipo de suelo no se puede cambiar. En el futuro, esta condición se puede abordar fácilmente proporcionando como entrada solo los atributos que se pueden cambiar.

Trabajo Futuro

Dada la gran cantidad de contrafácticos que el método ha logrado producir en la mayoría de los casos, queda pendiente abordar la multiplicidad contrafáctica explorando varios métodos de filtrado, poda y selección. Investigar técnicas de visualización y desarrollar una interfaz que facilite navegar la gran cantidad de contrafácticos y poder seleccionar los atributos modificables, y su preferencia ante otros, para ayudar a los usuarios a navegar por las explicaciones de salida y seleccionarlas en función de las preferencias (posiblemente implícitas) del usuario. Además, sería beneficioso realizar estudios de usuarios para anali-

zar la calidad percibida y el beneficio de nuestros contrafácticos para evitar "descuidar a los usuarios" [28].

Se nos ocurren diferentes métricas que pueden servir de guía para mejorar la precisión del modelo sustituto y mejorar la plausibilidad de los contrafácticos generados en general:

- *Convergencia*: Supervisar cómo cambia la precisión del modelo sustituto a través de las iteraciones; se debería ver que el error inicial es mayor y luego disminuye.
- *Calidad*: Calcular el porcentaje de contrafácticos plausibles entre los generados y los filtrados, esto determina cuantos contrafácticos "basura" se están generando de más que terminan siendo descartados.

Se podrían realizar además análisis para revisar la eficiencia de la metodología presentada, como oportunidades destacamos:

- *Análisis de ablación*; usar una búsqueda aleatoria de contrafácticos como linea de base - Random Search (RS), e ir agregandole la generación de vecinos - RS+Neighborhood (RSN), y luego con optimización bayesiana - RSN+Bayesian Optimization (RSN-BO), este ultimo seria equivalente al método presentado, BayCon.
- *Ejecutarlo con el conjunto de datos de MNIST*: la cual es una gran colección de dígitos escritos a mano. El objetivo sería generar contrafácticos con cambios en la imagen de los dígitos para que se parezca a otro dígito elegido por el usuario. Otro estudio similar es con algún modelo especializado en predecir objetos en imágenes, por ejemplo perros, y tratar de generar cambios mínimos en la imagen original para que deje de predecir que es un perro.
- *Validación*: BayCon es estocástico y no se puede garantizar la solución final óptima. Para hacer eso, BayCon necesita ser validado. Para ello se podría comparar a BayCon con métodos deterministas en conjuntos de datos mas chicos. Finalmente, también podría validarse en modelos de decisión reales para examinar la validez de las soluciones.

Bibliografía

[1] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[2] Francesca Bampa, Lilian O'Sullivan, Kirsten Madena, Taru Sandén, Heide Spiegel, Christian Bugge Henriksen, Bhim Bahadur Ghaley, Arwyn Jones, Jan Staes, Sylvain Sturel, et al. Harvesting european knowledge on soil functions and land management using multi-criteria decision analysis. *Soil use and management*, 35(1):6–20, 2019.

[3] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[4] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[5] Ruth MJ Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In *IJCAI*, pages 6276–6282, 2019.

[6] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[7] Mark William Craven. *Extracting comprehensible models from trained neural networks*. The University of Wisconsin-Madison, 1996.

[8] Zhicheng Cui, Wenlin Chen, Yujie He, and Yixin Chen. Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 179–188, 2015.

[9] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*, pages 448–469. Springer, 2020.

[10] Chao De Yu. What is the difference between traditional programming and machine learning? 2021.

[11] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[12] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018.

[13] Alberto Fernandez, Francisco Herrera, Oscar Cordon, Maria Jose del Jesus, and Francesco Marcelloni. Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to? *IEEE Computational intelligence magazine*, 14(1):69–81, 2019.

[14] Martin Gjoreski, Vladimir Kuzmanovski, and Marko Bohanec. Generating alternatives for dex models using bayesian optimization. 2020.

[15] Michael Gleicher. A framework for considering comprehensibility in modeling. *Big data*, 4(2):75–88, 2016.

[16] Ian Goodfellow, Nicolas Papernot, Patrick McDaniel, Reuben Feinman, Fartash Faghri, Alexander Matyasko, Karen Hambardzumyan, Yi-Lin Juang, Alexey Kurakin, Ryan Sheatsley, et al. cleverhans v0. 1: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 1, 2016.

[17] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

[18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[19] David Gunning. Explainable artificial intelligence (xai): technical report defense advanced research projects agency darpa-baa-16-53. *DARPA, Arlington, USA*, 2016.

[20] Fritz Heider. *The psychology of interpersonal relations*. Psychology Press, 2013.

[21] Germund Hesslow. The problem of causal selection. *Contemporary science and natural explanation: Commonsense conceptions of causality*, pages 11–32, 1988.

[22] Denis J Hilton. Mental models and causal explanation: Judgements of probable cause and explanatory relevance. *Thinking & Reasoning*, 2(4):273–308, 1996.

[23] Denis J Hilton and Ben R Slugoski. Knowledge-based causal attribution: The abnormal conditions focus model. *Psychological review*, 93(1):75, 1986.

[24] Robert Hoffman, Tim Miller, Shane T Mueller, Gary Klein, and William J Clancey. Explaining explanation, part 4: a deep dive on deep nets. *IEEE Intelligent Systems*, 33(3):87–95, 2018.

[25] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pages 507–523. Springer, 2011.

[26] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

[27] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *IJCAI*, pages 2855–2862, 2020.

[28] Mark T Keane, Eoin M Kenny, Eoin Delaney, and Barry Smyth. If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques. *arXiv preprint arXiv:2103.01035*, 2021.

[29] Michael T Lash, Qihang Lin, Nick Street, Jennifer G Robinson, and Jeffrey Ohlmann. Generalized inverse classification. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 162–170. SIAM, 2017.

[30] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv preprint arXiv:1907.09294*, 2019.

[31] David Lewis. Philosophical papers: Volume 2. 1987.

[32] Daniel James Lizotte. Practical bayesian optimization. 2008.

[33] Tania Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470, 2006.

[34] Bertram F Malle. *How the mind explains behavior: Folk explanations, meaning, and social interaction*. MIT press, 2006.

[35] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[36] Jonas Močkus. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pages 400–404. Springer, 1975.

[37] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[38] Jonathan Moore, Nils Hammerla, and Chris Watkins. Explaining deep learning models with constrained adversarial examples. In *Pacific Rim international conference on artificial intelligence*, pages 43–56. Springer, 2019.

[39] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[40] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020.

[41] Alun Preece, Dan Harborne, Dave Braines, Richard Tomsett, and Supriyo Chakraborty. Stakeholders in explainable ai. *arXiv preprint arXiv:1810.00184*, 2018.

[42] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[43] Piotr Romashov, Martin Gjoreski, Kacper Sokol, Maria Vanina Martinez, and Marc Langhenreich. Baycon: Model-agnostic bayesian conterfactual generator. In *IJCAI*, 2022.

[44] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[45] Chris Russell. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28, 2019.

[46] Jay Selig. What is machine learning? a definition. 2022.

[47] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pages 2171–2180. PMLR, 2015.

[48] Kacper Sokol, Alexander Hepburn, Rafael Poyiadzi, Matthew Clifford, Raul Santos-Rodriguez, and Peter Flach. Fat forensics: a python toolbox for implementing and deploying fairness, accountability and transparency algorithms in predictive systems. *Journal of Open Source Software*, 5(49):1904, 2020.

[49] Thomas Spooner, Danial Dervovic, Jason Long, Jon Shepard, Jiahao Chen, and Daniele Magazzeni. Counterfactual explanations for arbitrary regression models. *arXiv preprint arXiv:2106.15212*, 2021.

[50] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813, 2020.

[51] Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

[52] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

[53] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[54] Darrell M West. *The future of work: Robots, AI, and automation.* Brookings Institution Press, 2018.

[55] Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[56] Joseph Jay Williams, Tania Lombrozo, and Bob Rehder. The hazards of explanation: Overgeneralization in the face of exceptions. *Journal of Experimental Psychology: General*, 142(4):1006, 2013.