Resumen Rat Unidad 2

Master Machine Learning Algorithms

> ***Capítulo 3 "Algorithms Learn a Mapping from Input to Output"***

Es importante tener presente el error (e) cuando consideramos calcular

$$Y = f(X) + e$$

Donde f es la función que queremos aprender y e es el error que se comete al estimar f.

Learning a Function to Make Predictions

Asignarle valores a X para que la función f nos devuelva un valor Y, que permita realizar predicciones lo más cercanas a la realidad.

Techniques For Learn a Function

Existen diferentes formas de hacerlo, cada cual tiene diferentes asunciones y limitaciones, pueden ser lineales o no lineales.

Debemos evaluar entre las diferentes técnicas, cuál es la que mejor se ajusta a nuestro problema.

> ***Capítulo 4 "Parametric and Nonparametric Machine Learning Algorithms"***

- Los algoritmos paramétricos simplifican la función a una forma funcional conocida.
- Los algoritmos no paramétricos pueden aprender cualquier mapeo desde las entradas hasta las salidas.
- Todos los algoritmos se pueden organizar entre grupos paramétricos y no paramétricos.

Parametric Machine Learning Algorithms

Las asunciones pueden simplificar en gran medida el proceso de aprendizaje, pero puede incluso limitar lo que se puede aprender.

Los algoritmos que simplifican la función a una forma conocida se denominan algoritmos de aprendizaje automático paramétrico.

Los algoritmos involucran dos pasos:

1. Seleccionar un formulario para la función
2. Aprender los coeficientes de la función desde los datos de entrenamiento.

Una funcionalidad conocida es una función lineal, por ejemplo:

$$B0 + B1 * X1 + B2 * X2 = 0$$

Donde $B0$, $B1$ y $B2$ son los coeficientes que deben ser aprendidos a partir de los datos de entrenamiento. Y $X1$ y $X2$ son las variables de entrada. Suponiendo que el formulario funcional simplifica el proceso de aprendizaje. Ahora, todos necesitamos que estimen el coheficionte d de la ecuación y tenemos un modelo predictivo al problema.

Algunos ejemplos de algoritmos paramétricos son:

- Regresión logística
- Análisis discriminante lineal
- Perceptrón

Beneficios de Algoritmos para Aprendizaje automatizado

- Simplicidad
- Rapidez
- Menos información

Limitaciones de Algoritmos para Aprendizaje automatizado

- Restricciones:Al elegir una forma funcional, estos métodos quedan altamente limitados a la forma especificada.
- Complejidad Limitada: Estos métodos son más adecuados para problemas más simples.
- Ajuste Deficiente: En la práctica, es poco probable que estos métodos se ajusten correctamente a la función de mapeo subyacente.

Nonparametric Machine Learning Algorithms

Los algoritmos que no hacen suposiciones sobre la forma funcional son llamados algoritmos de aprendizaje automático no paramétricos.

Los métodos no paramétricos son útiles cuando se dispone de una gran cantidad de datos y no se cuenta con conocimiento previo, y cuando no se desea preocuparse demasiado por elegir exactamente las características adecuadas.

Algunos ejemplos de algoritmos no paramétricos son:

- Decision Trees como CART y C4.5
- Naive Bayes
- Support Vector Machines
- Neural Networks

Beneficios de Algoritmos para Aprendizaje automatizado

- Flexibilidad: Pueden aprender cualquier cosa.
- Potencia: No hay límite en lo que pueden aprender.
- Performance: Pueden ajustarse a los datos de entrenamiento.

Limitaciones de Algoritmos para Aprendizaje automatizado

- Más datos: Requieren una gran cantidad de datos de entrenamiento para estimar la función de mapeo de entrada a salida.
- Más lentos: Mucho más lento para entrenar, ya que a menudo tienen muchos más parámetros para entrenar.
- Sobreajuste: más riesgo de sobreajuste de los datos de entrenamiento y es más difícil explicar por qué se hacen predicciones específicas.

> ***Capítulo 5 "Supervised and Unsupervised Machine Learning Algorithms"***

Supervised Machine Learning

La Mayoría utiliza este tipo de ML, tomando las variables X y aprendiendo la funcion de mapeo de la entrada a la salida Y.

$$Y = f(X)$$

Clasificación: Un problema de clasificación es cuando la variable de salida es una categoría, como rojo o azul o enfermedad y no enfermedad. **Regresión:** Un problema de regresión es cuando la variable de salida es un valor real, como dólares o peso.

Algunos ejemplos de algoritmos de aprendizaje supervisado son:

- Regresión lineal para problemas de regresión.
- Bosques aleatorios para problemas de clasificación y regresión.
- Vectores de soporte para problemas de clasificación.

Unsupervised Machine Learning

Los algoritmos de aprendizaje automático no supervisados aprenden de los datos de entrada X, sin una variable de salida conocida.

$$X = f()$$

Los problemas de aprendizaje no supervisados se pueden agrupar en problemas de agrupación y asociación.

- **Clustering**: Identificar grupos inherentes en los datos, como categorizar clientes según sus patrones de compra.
- **Asociación**: Descubrir reglas que describen relaciones significativas en los datos, como la tendencia de las personas que compran A a también comprar B.

Algunos ejemplos de estos algoritmos son:

- K-Means para problemas de agrupación.
- Apriori para problemas de asociación.

Semi-supervised Machine Learning

Los algoritmos de aprendizaje automático semisupervisados aprenden de los datos de entrada X y de una variable de salida conocida Y.

Resumen

Supervisado: Todos los datos están etiquetados y los algoritmos aprenden a predecir la salida a partir de los datos de entrada. **No supervisado:** Todos los datos están sin etiquetar y los algoritmos aprenden la estructura inherente de los datos de entrada. **Semi-supervisado:** Algunos datos están etiquetados, pero la mayoría no lo está, y se pueden utilizar técnicas tanto supervisadas como no supervisadas en combinación.

> ***Capítulo 6 "The Bias-Variance Trade-Off"***

Después de leer este capítulo, sabrá:

- Los errores pueden ser descompuestos en error de sesgo y error de varianza.
- Los sesgos refieren a simplificar asunciones hechas por algoritmos para hacer el problema más fácil de resolver.
- La varianza se refiere a la sensibilidad de un modelo para cambiar los datos de entrenamiento.
- Las aplicaciones de ML por modelos predictivos son mejor entendidas a través del marco de sesgo-varianza.

Descripcion de sesgo y varianza

El error de predicción de un modelo se puede descomponer en 3 partes:

1. Error de sesgo
2. Error de varianza
3. Error irreducible

El *error irreducible* se puede deber a factores como el desconocimiento de la influencia del mapeo de las variables de entrada con las de salida. Por tanto nos centraremos en el error de sesgo y varianza.

Error de sesgo

Los sesgos son simplificaciones de asunciones hechas por un modelo para hacer la funcion objetivo más fácil de aprender. Generalmente los algoritmos parametricos tienen un alto sesgo haciendolos más rápidos de aprender y más faciles de entender, pero generalmente menos flexibles.

- **Bajo Sesgo:** sugieren más asunciones sobre el formulario de la funcion objetivo. Ej: *Decision trees, k-Nearest Neighbors y Support Vector Machines*.
- **Alto Sesgo:** sugieren menos asunciones sobre el formulario de la funcion objetivo. Ej: *Linear Regression, Linear Discriminant Analysis y Logistic Regression*.

Error de varianza

La varianza es la cantidad que la estimación de la funcion objetivo cambiará si se usa diferentes datos de entrenamiento. La función objetivo es estimada por la información de entrenamiento por un algoritmo de ML, entonces debemos esperar que el algoritmo tenga alguna varianza.

Si el algoritmo es bueno este sacaría el subyacente mapeo de las variables de entrada a las de salida. Los algoritmos de ML con alta varianza son fuertemente influenciados por las especificaciones de los datos de entrenamiento. Esto significa que la espeficiación de los entrenamientos influencian el numero y tipo de parametros usados para caracterizar la funcion mapeo.

- **Baja Varianza:** sugiere pequenos cambios de estimación de la función objetivo con cambios del conjunto de datos de entrenamiento. Ej: *Linear Regression, Linear Discriminant Analysis y Logistic Regression*.
- **Alta Varianza:** sugiere grandes cambios de estimación de la función objetivo con cambios del conjunto de datos de entrenamiento. Ej: *Decision trees, k-Nearest Neighbors y Support Vector Machines*.

Compensación de sesgo y varianza

La meta de cualquier algoritmo de ML supervisado es lograr bajo sesgo y baja varianza. En cambio el algoritmo debe alcanzar buena performance de predicción.

- Algoritmos paramétricos o lineales de ML aveces tienen algo sesgo pero baja varianza.
- Algoritmos no paramétricos o no lineales de ML aveces tienen algo varianza pero bajo sesgo.

La parametrización de algoritmos de ML aveces es una batalla sin balance entre sesgo y varianza. Algunos ejemplos:

- El Algoritmo k-Nearest Neighbors tiene bajo sesgo y alta varianza, pero la compensación puede cambiar por incremento del valor de k cuando incrementa el numero de neighbors que contribuyen a la predicción t y en cambio incrementan el sesgo del modelo.
- El Algoritmo Support Vector Machines tiene bajo sesgo y alta varianza, pero la compensación puede cambiar por incremento del valor de C influencia el numero de violaciones de margen permitido en el dato de entrenamiento que incrementa el sesgo pero disminuye la varianza.

Existe un equilibrio entre estas preocupaciones y los algoritmos seleccionados, junto con su configuración, encuentran diversos equilibrios para abordar este balance en el problema. Aunque no podemos calcular los errores de varianza y sesgo reales debido a la falta de conocimiento sobre la función objetivo subyacente, estos términos proporcionan un marco para comprender cómo los algoritmos de aprendizaje automático se comportan en su búsqueda de rendimiento predictivo.

> ***Capítulo 7 "Overfitting and Underfitting"***

La causa del rendimiento deficiente en el aprendizaje automático es el ajuste excesivo o insuficiente de los datos.

- Ese sobreajuste se refiere a aprender demasiado bien los datos de entrenamiento a expensas de no generalizar bien a datos nuevos.
- Ese subajuste se refiere a falla al aprender suficientemente el problema de los datos de entrenamiento.
- El sobreajuste es el problema más común de hecho y puede abordarse mediante el uso de métodos de re-muestreo y un conjunto de verificación retenido.

Generalización en ML

En ML describimos la funcion objetivo de aprendizaje desde datos de entrenamiento como aprendizaje inductivo. Inducción refiere a aprender conceptos generales desde ejemplos especificos que es exactamente el problema que el ML supervisado ayuda a resolver. Generalización refiere a que tan bien los conceptos aprendidos por un modelo de ML aplican a ejemplos no vistos cuando el modelo estaba siendo entrenado. La meta de un buen modelo de ML es generalizar bien desde los datos de entrenamiento hacia cualquier dato de un problema dominio.

Ajuste estadístico

En estadistica se refiere a que tan bien se aproxima a la función objetivo. Este es un buen término para usar en el aprendizaje automático, ya que los algoritmos de aprendizaje automático supervisados buscan aproximar la función de mapeo subyacente desconocida de las variables de salida dadas las variables de entrada.

Sobreajuste en ML

El sobre ajuste se da cuando cargamos demasiado el modelo de aprendizaje con detalles, que afecta el rendimiento negativamente. Afecta negativamente en la habilidad del modelo para generalizar desde los datos de entrenamiento a datos nuevos. El sobreajuste es más parecido con modelos no parametricos y no lineales, que tienen más flexibilidad cuando estan aprendiendo la función objetivo. Muchos algoritmos de ML no parametricos incluyen parametros o tecnicas que limitan y restringen que tan detallado aprende el modelo de aprendizaje. Ej: decision trees son algoritmos de ML no parametricos que es muy flexible y sujeto a sobreajuste de datos de entrenamiento. Este problema puede abordarse podando un árbol después de que ha aprendido, con el fin de eliminar algunos de los detalles que ha capturado.

Subajuste en ML

El subajuste se refiere a un modelo que no puede modelar los datos de entrenamiento ni generalizar a nuevos datos. Este tipo de modelo tiene un rendimiento deficiente en los datos de entrenamiento y no es adecuado. Aunque no se discute mucho, se puede detectar fácilmente mediante métricas de rendimiento. La solución es probar otros algoritmos de aprendizaje automático, y contrasta con el problema del sobreajuste.

Un buen ajuste en ML

El buen ajuste se refiere a un modelo que captura la esencia subyacente de los datos de entrenamiento, pero puede generalizar bien a nuevos datos. Un buen ajuste es el objetivo de cualquier modelo de aprendizaje automático supervisado. Aunque no se discute mucho, se puede detectar fácilmente mediante métricas de rendimiento. La solución es probar otros algoritmos de aprendizaje automático, y contrasta con los problemas de sobreajuste y subajuste.

Como limitar el sobreajuste

Tanto sobreajuste como subajuste pueden llevar a un pobre modelo de rendimiento. Pero el sobreajuste es el problema más común. Hay dos tipos de técnicas que se pueden aplicar para limitar el sobreajuste:

1. Usar una técnica de remuestreo para estimar la certeza del modelo.
2. Mantener un conjunto de datos de validación.

La técnica maás popular para remuestreo es k-fold cross-validation. Este te permite entrenar y probar tu modelo k-veces en diferentes subconjuntos de datos de entrenamiento y construir una estimación del rendimiento del modelo de ML en datos no vistos. Una validacón del conjunto de datos es simplemente un subconjunto de tus datos de entrenamiento que mantienes desde tus algoritmos de ML hasta el final mismo del proyecto. Despues que hayas seleccionado y modificado tus algoritmos de Ml con conjuntos de datos de entrenamiento para evaluar los modelos aprendidos. Utilizar la validación cruzada es un estándar de oro en el aprendizaje automático aplicado para estimar la precisión del modelo en datos no vistos. Si tienes los datos, utilizar un conjunto de validación también es una práctica excelente.