Aprendizaje de Máquinas, Laboratorio #1

Apunte teórico: Árboles de Decisión, Naïve Bayes

1. Árboles de decisión

El aprendizaje de árboles de decisión es utilizado para problemas de clasificación cuya función objetivo son valores discretos o categóricos. Los árboles aprendidos pueden ser representados como conjuntos de reglas *si-entonces* siendo más legibles para las personas, similares a las utilizadas en varios lenguajes de programación (if, then, else). Este método de aprendizaje es uno de los algoritmos más populares de inferencia inductiva. Ha sido aplicado en un amplio rango de tareas de aprendizaje como por ej. diagnóstico médico o asesoramiento de riesgo crediticio.[1]

1.1. Representación de árboles de decisión

Los árboles de decisión clasifican las instancias, ordenándo sus pares atributos-valor y recorriendo el árbol hasta llegar al nodo hoja que contiene el resultado de la clasificación. Por ejemplo, dado el árbol de decisión de la Fig. 1 la instancia <*Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong*> resultará en *PlayTennis=NO*.

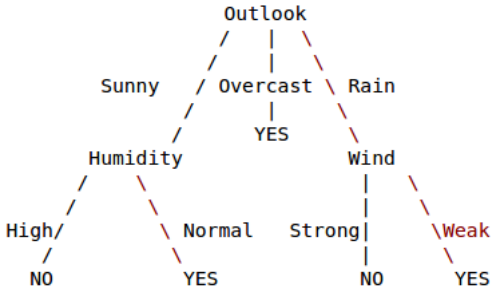

Figura 1: Un árbol de decisión para el concepto *PlayTennis*. Este árbol de decisión determina de acuerdo al clima de un sábado a la mañana si jugar tenis o no. Un nuevo ejemplo o instancia es clasificado, recorriendo el árbol desde la raíz (Outlook) en base a los valores de los atributos que otorga el ejemplo, hasta llegar a un nodo hoja (en este caso YES o NO)

1.2. Problemas apropiados para árboles de decisión

- *Las instancias son representadas por pares atributo-valor*: las instancias son descriptas por un conjunto de atributos, por ej. Temperature, y sus valores, ej. Hot.
- *La función objetivo tiene valores de salida discretos.* El árbol de decisión de la Fig. 1 asigna el valor YES o NO a cada ejemplo, resultando en un clasificador binario. Fácilmente puede extenderse a funciones de aprendizaje con más valores de salida.

[1] Los contenidos de la presente sección han sido desarrollados tomando como referencia principal el capítulo 3 de *Machine Learning* de Tom Mitchell [1]

- *Representación: los árboles de decisión representan una disyunción de conjunciones*, por ejemplo, el árbol de decisión de la Fig. 1 corresponde con la expresión
(Outlook=Sunny AND Humidity=Normal) OR (Outlook=Overcast) OR (Outlook=Rain AND Wind=Weak)

- *Tolerancia a errores*: estos métodos son bastante robustos a errores en los datos del conjunto de entrenamiento.

- *Valores faltantes*: Los árboles de decisión pueden ser usados en casos que algunos valores de atributos de ejemplos del conjunto de entrenamiento sean desconocidos. Ver sección 3.7.3 de [1].

1.3. El algoritmo básico de aprendizaje de árboles de decisión

Existen varios algoritmos que aprenden árboles de decisión empleando un enfoque *top-down* y de búsqueda voraz a través de todo el espacio posible de árboles de decisión. Este enfoque es ejemplificado por ID3 (inductive decision tree) desarrollado por Quinlan en 1986, que usaremos como nuestro algoritmo básico para construir el árbol de decisión. El algoritmo ID3 aprende árboles de decisión construyéndolos desde arriba hacia abajo (enfoque *top-down*), empezando por responder la pregunta *¿cuál es el atributo que debe ser la raíz del árbol?*. Para responder esta pregunta, cada instancia de atributo es evaluada usando un test estadístico que determine cuan bueno es clasificando los ejemplos de entrenamiento. El mejor atributo es seleccionado y usado como nodo raíz del árbol. Se crea una rama por cada valor posible del atributo elegido como nodo raíz. Por cada rama, se elige otro atributo en base al test estadístico. El proceso es repetido sucesivamente con los ejemplos de entrenamiento asociados con cada nodo descendiente. Este esquema es una búsqueda voraz para formar un árbol de decisión, en el cual el algoritmo nunca reconsidera decisiones pasadas volviendo a evaluar nodos anteriores. Una versión simplificada del algoritmo es descripta en 1.3.3.

La decisión central del algoritmo ID3 es elegir cuál es el atributo que será nodo del árbol. Nos gustaría elegir el atributo que sea más útil para clasificar los ejemplos del conjunto de entrenamiento. Definimos una propiedad estadística, llamada *Ganancia de la información*, que mide cuan bueno es un atributo dado para separar ejemplos del conjunto de entrenamiento de acuerdo a su clasificación o atributo de clase (función objetivo, target attribute, clase). ID3 usa *Ganancia de la información* como test estadístico para elegir entre varios atributos cual es el candidado en cada paso mientras construye el árbol.

1.3.1. Entropía

Para poder definir la *Ganancia de la información* es preciso que empecemos definiendo una medida muy utilizada en teoría de la información: la Sra *Entropía*. La Entropía caracteriza la impureza de una colección de ejemplos. Dado un conjunto S, que contiene ejemplos clasificados como positivos y negativos, la entropía de S respecto a esta clasificación booleana es

$$Entropy(S) = -p_+ log_2 p_+ - p_- log_2 p_- \tag{1}$$

donde p_+ es la proporción de ejemplos positivos en S y p_- es la proporción de ejemplos negativos en S. Para todos los cálculos que involucren entropía, definiremos que $0 * log(0) = 0$. Note que la entropía es cero si todos los elementos de S pertenecen a la misma clase, por ej. si todos los miembros son positivos o todos son negativos. La entropía es 1 cuando una colección de ejemplos contiene el mismo número de ejemplos clasificados en positivo y negativo. Los valores de entropia van de 0 a 1. Si la cantidad de ejemplos negativos y positivos difiere, la entropía dará un valor entre 0 a 1.

Cuadro 1: Ejemplos de entrenamiento para PlayTennis

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	NO
D2	Sunny	Hot	High	Strong	NO
D3	Overcast	Hot	High	Weak	YES
D4	Rain	Mild	High	Weak	YES
D5	Rain	Cold	Normal	Weak	YES
D6	Rain	Cold	Normal	Strong	NO
D7	Overcast	Cold	Normal	Strong	YES
D8	Sunny	Mild	High	Weak	NO
D9	Sunny	Cold	Normal	Weak	YES
D10	Rain	Mild	Normal	Weak	YES
D11	Sunny	Mild	Normal	Strong	YES
D12	Overcast	Mild	High	Strong	YES
D13	Overcast	Hot	Normal	Weak	YES
D14	Rain	Mild	High	Strong	NO

Para tener un ejemplo, tome lápiz y papel, calcule la entropía del los ejemplos de entrenamiento dados en la Tabla 1. Si comprendió bien como aplicar la fórmula el resultado le dará 0.94.

1.3.2. Ganancia de información

La ganancia de información de un atributo A respecto a un conjunto de ejemplos S, se define como:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} * Entropy(S_v) \quad (2)$$

donde $Values(A)$ es el conjunto posible de valores que toma el atributo A, S_v es subconjunto de S en cuyos ejemplos el atributo A toma el valor v y S el conjunto total de ejemplos. La entropía descripta en el segundo término es la suma de las entropías de cada subconjunto S_v. $Gain(S, A)$ es la reducción de la entropía causada por conocer el valor de un atributo. Nuevamente lo invitamos a tomar lápiz y papel para calcular la ganancia de información de cada uno de los atributos del conjunto S dado en la tabla 1. Los resultados son: $Gain(S, Outlook) = 0{,}246$; $Gain(S, Humidity) = 0{,}151$; $Gain(S, Wind) = 0{,}048$; $Gain(S, Temperature) = 0{,}029$

1.3.3. ID3

El seudocódigo de ejecución del algoritmo ID3 puede encontrarse en tabla 3.1 de [1].

2. Aprendizaje Bayesiano

Los métodos de *aprendizaje Bayesiano* son altamente relevantes para nuestro estudio del aprendizaje de máquinas por dos diferentes razones. Primero, los algoritmos que aprenden calculando explicitamente

las probabilidades de las distintas hipótesis, como es el caso del clasificador *naïve* Bayes, son uno de los enfoques más practicos para los algoritmos de clasificación, con performances comparativas a algoritmos más sofisticados como los son los *árboles de decisión* o las *redes neuronales artificiales*. Discutiremos este algoritmo en esta sección.[2]

La segunda razón por la cual los métodos Bayesianos son importantes en nuestro estudio del aprendizaje automático es que proveen una perspectiva útil para comprender muchos de los algoritmos que manipulan probabilidades explícitamente. En particular, discutiremos el uso del *error cuadrático medio* y *cross entropy* como funciones de pérdida/error en el caso de redes neuronales (omitido). Así mismo, este formalismo provee fundamentos teóricos de un importante bias inductivo: *Occam's razor*. Aunque omitido en el presente documento, la referencia principal [1] discute este bias inductivo para el caso de árboles de decisión que favorece árboles pequeños, argumentando su equivalencia con el principio de *minimum description length*.

Las carácteristicas del aprendizaje Bayesiano más relevantes para nuestro estudio incluye:

- La observación de cada ejemplo de entrenamiento puede incrementar o reducir nuestra estimación de la probabilidad de que cierta hipótesis sea correcta. Esto provee un enfoque más flexible a los algoritmos que directamente descartan hipótesis cuando estas contradicen aunque sea un ejemplo.
- Conocimiento previo sobre las hipótesis (e.g., cuales son más probables previo a observar los datos), y sobre los datos, puede combinarse con la observación de los datos en la determinación de la probabilidad final de una hipótesis.
- La clasificación de nuevas instancias puede resultar de la combinación de múltiples hipótesis, pesadas por sus probabilidades.
- Incluso cuando el aprendizaje Bayesiano es intratable computacionalmente, puede utilizarse como un estandar para comparar la performance de otros métodos más practicos.

Una de las principales dificultades de estos métodos es el alto costo computacional necesario para determinar la hipótesis óptima, que por lo general es lineal al número de hipótesis. En algunos casos especiales sin embargo, este costo puede reducirse significativamente, tal como es el caso del algoritmo de naïve Bayes.

Comenzaremos discutiendo algunos aspectos formales del aprendizaje Bayesiano como es el *teorema de Bayes*, y los principios de *máxima verosimilitud* (*maximum likelihood*), y *máximo a posteriori*, aplicados al caso de estimación de hipótesis. Discutiremos luego como aplicar este formalismo para justificar el uso del error cuadrático medio y cross-entropy como funciones de pérdida en redes neuronales, y concluimos presentando el algoritmo de naïve Bayes. (Omitido: redes Bayesianas y el algoritmo EM).

2.1. Teorema de Bayes

El aprendizaje inductivo consiste en el problema de encontrar la hipótesis h perteneciente al espacio de hipótesis $\mathcal{H}$ que mejor explica los datos del conjunto de entrenamiento $\mathcal{T}$. Probabilísticamente, podemos cuantificar que tan bien una hipótesis explica los datos a través de la probabilidad de la hipótesis dado que se ha observado $\mathcal{T}$, i.e., la probabilidad *condicional* $\Pr(h \mid \mathcal{T})$. A esta probabilidad condicional se la denomina *probabilidad posterior*, ya que es la probabilidad luego de haber observado los datos. De esta manera, el problema de aprendizaje inductivo puede expresarse probabilísticamente como el

[2] Los contenidos de la presente sección han sido desarrollados tomando como referencia principal el capítulo 6 de *Machine Learning* de Tom Mitchell [1]

problema de encontrar la hipótesis h_{MAP} con máxima posterior respecto al conjunto de entrenamiento $\mathcal{T}$, es decir,

$$h_{MAP} = \arg\max_{h \in \mathcal{H}} \Pr(h \mid \mathcal{T}). \tag{3}$$

Como uno puede intuir, no es nada claro como computar la probabilidad posterior. Por el contrario, conocida la hipótesis, pareciera ser una tarea mucho más sencilla computar la probabilidad de los datos, i.e., $\Pr(\mathcal{T} \mid h)$. A esta probabilidad se la llama *verosimilitud de los datos* (*data likelihood* en inglés). Veamos un ejemplo:

Ejemplo 1. *Consideremos por ejemplo el caso de diagnóstico médico donde tenemos dos posibles hipótesis: que el paciente tiene o no tiene cancer. Además, los datos proveidos corresponden al resultado de un test de cierto laboratorio X, que pueden tomar los valores* positivo *(simbolicamente $+$) o* negativo *(simbolicamente $-$). El problema consiste en determinar la posterior* $\Pr(cancer \mid test)$, $test \in \{+, -\}$. *Si bien no es claro como obtener este valor, si es esperable que el laboratario X haya reportado la calidad de su test, es decir, la verosimilitud* $\Pr(+ \mid cancer)$ *y* $\Pr(- \mid \neg cancer)$, *que indica cual es la probabilidad de que el test resulte positivo cuando el paciente tiene cancer, y negativo cuando no lo tiene.*

El teorema de Bayes nos provee un formalismo para computar la posterior a partir de la verosimilitud y las probabilidades $\Pr(D)$ y $\Pr(h)$:

Teorema 1. *Teorema de Bayes*

$$\Pr(h \mid \mathcal{T}) = \frac{\Pr(\mathcal{T} \mid h) \Pr(h)}{\Pr(D)}$$

donde $\Pr(D)$ modela la probabilidad de haber observardo esta muestra de ejemplos en particular, y la $\Pr(h)$, llamada *probabilidad prior*, modela algún conocimiento previo (al observar los datos) que tengamos sobre cual es la hipótesis correcta.

La demostración del teorema parte de la definición de probabilidad condicional, donde $\Pr(h \mid \mathcal{T}) = \frac{\Pr(h,\mathcal{T})}{\Pr(\mathcal{T})}$ y $\Pr(\mathcal{T} \mid h) = \frac{\Pr(h,\mathcal{T})}{\Pr(h)}$, y concluye despejando la conjunta $\Pr(h, \mathcal{T})$, igualando, y despejando $\Pr(h \mid \mathcal{T})$.

Es interesante analizar el teorema, observando que como es esperado intuitivamente, la posterior $\Pr(h \mid \mathcal{T})$ crece con $\Pr(h)$ y $\Pr(\mathcal{T} \mid h)$. También que, la posterior decrece cuando $\Pr(D)$ crece, porque cuanto más probable es que $\mathcal{T}$ sea observado independientemente de h, menor es la evidencia que $\mathcal{T}$ provee en favor de h.

Estamos en condiciones ahora de re-expresar el aprendizaje MAP de la Eq. (3) de la siguiente manera

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in \mathcal{H}} \Pr(h \mid \mathcal{T}) \\ &= \arg\max_{h \in \mathcal{H}} \frac{\Pr(\mathcal{T} \mid h) \Pr(h)}{\Pr(D)} \\ &= \arg\max_{h \in \mathcal{H}} \Pr(\mathcal{T} \mid h) \Pr(h) \end{aligned}$$

donde en el último paso se omitió $\Pr(D)$ ya que no depende de h, y por lo tanto no afecta la maximización.

Apliquemos este principio al ejemplo del cancer:

Ejemplo 1 (continuación) *Supongamos que se nos presenta un paciente al cual el test le dió positivo. ¿Deberíamos concluir que el paciente tiene cáncer? Es decir, por el principio MAP, ¿*$\Pr(cancer \mid +) > \Pr(\neg cancer \mid +)$*?. Para responder computamos estas probabilidades usando el teorema de Bayes. Para ello, necesitaremos el prior* $\Pr(cancer)$ *y la verosimilitud.. Como ya vimos, el laboratorio* X *nos provee la calidad de los tests, concretamente, nos dice que* $\Pr(+ \mid cancer) = 0{,}98$*, y* $\Pr(- \mid \neg cancer) = 0{,}97$*. Siendo cada una de estas condicionales una probabilidades por si misma, deben sumar* 1*, y por lo tanto* $\Pr(- \mid cancer) = 0{,}02$ *y* $\Pr(+ \mid \neg cancer) = 0{,}03$*. Además, de organismos públicos de estadísticas nos enteramos que tan solo un factor* 0,008 *de la población tiene cáncer, i.e.,* $\Pr(cancer) = 0{,}008$*. Con todo esto podemos calcular las posteriors:*

$$\Pr(cancer \mid +) \propto \Pr(+ \mid cancer)\Pr(cancer) = (0{,}98)0{,}008 = 0{,}0078$$
$$\Pr(\neg cancer \mid +) \propto \Pr(+ \mid \neg cancer)\Pr(\neg cancer) = (0{,}03)0{,}992 = 0{,}0298.$$

con lo que podemos concluir que a pesar de haber dado el test positivo, es mas probable que el paciente no tenga cáncer.

En la práctica suele considerarse una aproximación al aprendizaje MAP asumiendo que la prior $\Pr(h)$ es uniforme, i.e., antes de ver los datos no tenemos indicios de que ninguna de las hipótesis sea mejor que otra. En este caso maximizar la posterior resulta equivalente a maximizar la verosimilitud y por lo tanto nos referimos a este caso como *maxima verosimilitud* obteniendo

$$h_{ML} = \arg\max_{h \in \mathcal{H}} \Pr(\mathcal{T} \mid h)$$

donde las siglas en el subindice se tomaron del nombre en inglés (maximum likelihood).

2.2. Clasificadores Bayesianos

Hasta ahora hemos considerado como utilizar el teorema de Bayes para aprender hipótesis $h \in \mathcal{H}$, simplemente reportando la hipótesis h_{MAP} con máxima posterior, i.e., $h_{MAP} = \arg\max_{h \in \mathcal{H}} \Pr(h \mid \mathcal{T})$.

Podemos entonces aplicar el principio MAP para clasificar simplemente evaluando la hipótesis para cierto input $\mathbf{x}$, i.e., reportamos $h_{MAP}(\mathbf{x})$ como la clase mas probable de $\mathbf{x}$.

Esta no es sin embargo la manera ideal, ya que la información de hipótesis no óptimas puede ser de utilidad. Lo que nos lleva al concepto de *clasificador Bayesiano óptimo.*

2.2.1. Clasificador Bayesiano óptimo

A modo de ejemplo, consideremos un espacio de hipótesis con tres hipótesis h_1, h_2, y h_3, y supongamos que las posteriors de cada una de ellas son 0,3, 0,3, y 0,3, respectivamente. Tenemos entonces que $h_{MAP} = h_1$, y toda nueva instancia $\mathbf{x}$ entonces deberá ser clasificada de acuerdo a $h_1(\mathbf{x})$. Supongamos sin embargo que es el caso que h_1 clasificia a $\mathbf{x}$ como positiva, mientras que h_2 y h_3, ambas, clasifican a $\mathbf{x}$ como negativa. Vemos entonces que considerando todas las hipótesis, la probabilidad de clasificar negativo es 0,6 y positivo tan solo 0,4, contrario a lo indicado por el enfoque MAP.

Podemos formalizar esta idea de clasificación MAP, en lo que se llama el *clasificador Bayesiano óptimo*, planteando el problema directamente como un problema de maximización de la posterior del valor de la variable de clase dados los datos, a diferencia de maximización de posterior de la hipótesis. Formalmente, si la clasificación del nuevo ejemplo $\mathbf{x}$ puede tomar el cualquier valor v de algún conjunto

V de valores, buscamos el valor v_{MAP} que satisface

$$\begin{aligned} v_{MAP} &= \arg\max_{v \in V} \Pr(v \mid \mathcal{T}) = \\ &= \arg\max_{v \in V} \sum_{h \in} \Pr(v \mid h) \Pr(h \mid D) \end{aligned}$$

donde en el segundo paso aplicamos la ley de probabilidad total sobre la variable h, y tuvimos en cuenta que $\Pr(v \mid h, \mathcal{T}) = \Pr(v \mid h)$ ya que, una ves conocida la hipótesis h, los datos no nos proveen información relevante sobre la clasificación.

Continuando con el ejemplo, denotamos a la clasificación positiva como $+$ (y a la negativa por $-$), y notando que

$$\begin{array}{lll} \Pr(h_1 \mid \mathcal{T}) = 0{,}4, & \Pr(- \mid h_1) = 0, & \Pr(+ \mid h_1) = 1 \\ \Pr(h_2 \mid \mathcal{T}) = 0{,}3, & \Pr(- \mid h_2) = 1, & \Pr(+ \mid h_2) = 0 \\ \Pr(h_3 \mid \mathcal{T}) = 0{,}3, & \Pr(- \mid h_3) = 1, & \Pr(+ \mid h_1) = 0 \end{array}$$

tenemos que

$$\begin{aligned} \Pr(+ \mid \mathcal{T}) &= \sum_{h_i \in H} \Pr(+ \mid h_i) \Pr(h_i \mid \mathcal{T}) = 0{,}4 \\ \Pr(- \mid \mathcal{T}) &= \sum_{h_i \in H} \Pr(- \mid h_i) \Pr(h_i \mid \mathcal{T}) = 0{,}6. \end{aligned}$$

y por lo tanto $v_{MAP} = -$.

Lamentablemente, a pesar de arrojar en promedio este clasificador la mejor clasificación posible, este enfoque no es práctico ya que para encontrar v_{MAP}, debemos recorrer el espacio de hipótesis $\mathcal{H}$ completo.

Una alternativa es considerar otro enfoque para computar la posterior $\Pr(v \mid \mathcal{T})$ del valor de clase aplicando directamente el teorema de Bayes. Vemos esto en la siguiente sección

2.2.2. Clasificador Naïve Bayes

El problema de clasificación consiste en determinar el valor de clase v de cierto input $\mathbf{x}$, dado un conjunto de entrenamiento $\mathcal{T}$. El enfoque alternativo presentado aquí entonces hace explícito el hecho de que estamos clasificando el input $\mathbf{x}$ computando el valor v_{MAP} que maximiza la posterior $\Pr(v \mid \mathcal{T}, \mathbf{x})$. Aplicando Bayes tenemos entonces

$$\begin{aligned} v_{MAP} &= \arg\max_{v \in V} \Pr(v \mid \mathcal{T}, x) \\ &= \arg\max_{v \in V} \frac{\Pr(\mathcal{T}, x \mid v) \Pr(v)}{\Pr(\mathcal{T})} \\ &= \arg\max_{v \in V} \frac{\Pr(x \mid v, \mathcal{T}) \Pr(\mathcal{T} \mid v) \Pr(v)}{\Pr(\mathcal{T})} \\ &= \arg\max_{v \in V} \Pr(x \mid v, \mathcal{T}) \Pr(v) \end{aligned}$$

donde se aproximó $\Pr(\mathcal{T} \mid v)$ por $\Pr(\mathcal{T})$ por considerar de que el hecho de enterarse cual es la clasificación del input $\mathbf{x}$ no afecta en gran medida la probabilidad de que la muestra de entrenamiento sea exactamente $\mathcal{T}$.

Al clasificador resultante se le denomina *clasificador Bayesiano*. Para computarlo, nos queda computar $\Pr(\mathbf{x} \mid v, \mathcal{T})$, lo cual es posible hacer directamente del conjunto de entrenamiento $\mathcal{T}$ contabilizando, de entre los datapoints etiquetados con la clase v, la proporción de datapoints con valor $\mathbf{x}$. Si bien factible en teoría, esto no es practico ya que para una buena estimación de la frecuencia relativa, el dataset de entrenamiento debe tener muchos ejemplos para cada posible configuración de $\mathbf{x}$, lo cual claramente no es factible dado que existe un número exponencial de tales configuraciones (e.g., para inputs binarios, existen 2^D tales configuraciones).

Para resolver este problema, aplicamos la regla de la cadena sobre las variables $x_1, \ldots, x_D$ correspondientes a las componentes de $\mathbf{x}$, obteniendo

$$\Pr(\mathbf{x} \mid v, \mathcal{T}) = \prod_{i_1}^{D} \Pr(x_i \mid v, \mathcal{T}, x_{1:i-1}).$$

Esta expresión podría simplificarse de conocerse ciertas independencias entre las variables de entrada, condicionadas en v ya que podrian eliminarse del condicionante, reduciendo drasticamente el número de configuraciones. Las *redes Bayesianas* proponen un formalismo para aplicar esta idea sistematicamente, pero escapa al alcance de este escrito.

Presentamos sin embargo aquí el caso especial considerado por el algoritmo de naïve Bayes, el cual considera la aproximación un tanto drástia de que cada variable de entrada es independiente de todas las otras variables de entrada dada la clase. Por ser inocente suponer que esto se cumpla en la práctica se le dá el nombre de *naïve* (inocente en inglés). Esta aproximación indica entonces que $\Pr(x_i \mid v, \mathcal{T}, x_{1:i-1}) = \Pr(x_i \mid v, \mathcal{T})$, resultando en la clasificación dada por

$$v_{NB} = \arg\max_{v \in V} \Pr(\mathbf{x} \mid v, \mathcal{T}) \Pr(v) = \prod_{i_1}^{D} \Pr(x_i \mid v, \mathcal{T}) \Pr(v).$$

Vemos ahora que puede estimarse las cantidades $\Pr(x_i \mid v, \mathcal{T})$ del conjunto de entrenamiento sin perdida de calidad simplemente estimando, de entre el conjunto de datapoints que toman la clase v, la proporción de veces que la i-esima variable toma el valor x_i.

Computacionalmente este algoritmo presenta una enorme ventaja ya que no requiere realizar una búsqueda en el espacio de hipótesis. Además, a pesar de su simpleza, ha demostrado ser altamente competitivo con otros clasificadores, obteniendo en muchos casos resultados similares e incluso mejores.

Referencias

[1] Tom Mitchell. *Machine Learning*. McGRAW-HILL, 1997.