Teoría del Aprendizaje Estadístico

Nicolas Silva Nash
Departamento de Matemática
Universidad Nacional del Comahue

8 de enero de 2025

1. Introducción

La Teoría del Aprendizaje Estadístico proporciona la base teórica para muchos de los algoritmos de aprendizaje automático actuales y, sin lugar a dudas, es una de las ramas más bellamente desarrolladas de la inteligencia artificial en general. Nació con el perceptrón de Rossenblat y la escuela matemática de la Unión Soviética en la década de 1960, y ganó amplia popularidad en la década de 1990 tras el desarrollo de las llamadas Máquinas de Vectores de Soporte (*SVM*, por sus siglas en inglés), que se han convertido en una herramienta estándar para el reconocimiento de patrones en muchas disciplinas, que van desde la visión por computadora hasta la biología computacional.

Proporcionar la base para nuevos algoritmos de aprendizaje no ha sido la única motivación para desarrollar la Teoría del Aprendizaje Estadístico. También ha sido una gesta de caracter filosófico, en el intento de responder a la pregunta de qué nos permite extraer conclusiones válidas a partir de datos empíricos.

2. El aprendizaje

En este contexto, el *aprendizaje* es el proceso a través del cual pueden inferirse reglas generales a partir de ejemplos. Nos interesa entender como una máquina -una computadora- puede resolver ciertos problemas sin conocer las reglas de antemano, solo a partir de ejemplos y a través de un *algoritmo de aprendizaje*. El objetivo es que la máquina pueda no solo aprender a reconocer las reglas que rigen a los ejemplos dados, si no que también pueden generalizar dichas reglas para ejemplos que le serán presentados con posterioridad.

Llamamos a esta disciplina *aprendizaje automático* (en inglés, *Machine Learning*, literalmente "aprendizaje de máquinas") y reconocemos sus raíces en otras disciplinas: Estadística Matemática, Ciencias de la Computación e Inteligencia Artificial. Si bien el aprendizaje suele ser una parte fundamental de la mayoría de los esfuerzos en materia de Inteligencia Artificial, el objetivo del Machine

Learning es más acotado que el de su rama madre: En vez de intentar definir, explicar o generar comportamiento inteligente o *inteligencia*, aquí nos interesa solamente descubrir los mecanismos a través de los cuales las computadoras pueden resolver algunas tareas acotadas y bien definidas, y que en general escapan a soluciones que pueden ser especificadas con una cantidad finita de código de programación (reglas determinísticas).

Con fines ilustrativos, nos centraremos primero en el más conocido de los problemas del Machine Learning, el de clasificación. Consideremos dos espacios de variables: X, llamado *espacio de entrada*, e Y, el *espacio de etiquetas*. En un problema de clasificación, deseamos poder etiquetar correctamente elementos de X con los valores de Y. Por ejemplo, podríamos querer clasificar un conjunto de datos, en alguna representación fija, de distintos objetos en una cantidad de etiquetas como: silla, cama, microondas, perro, gato. Si esta representación es en imágenes de $N \times M$ pixeles en blanco y negro (realmente, matrices de orden $N \times M$ con coeficientes reales en el interavalo $[0, 1]$ que representan la intensidad de cada pixel, del negro al blanco), el espacio X es el conjunto de dichas matrices y el espacio Y las categorías distintas que corresponden a lo que las imágenes muestran. Con el fin de aprender, a un algoritmo se le muestran ejemplos de imágenes y sus respectivas etiquetas (X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n), a partir de los cuales este debe encontrar una función $f : X \to Y$, que comete la menor cantidad de errores posibles. A esta función f la llamamos *clasificador*.

3. La historia del aprendizaje automático

El primer modelo de aprendizaje automático, según Vapnik en [2], fue sugerido por F. Rosenblatt, un psicólogo estadounidense, al que llamó *perceptrón*, y su introducción constituye el comienzo del análisis matemático del aprendizaje. Conceptualmente, la idea del perceptrón no era nueva, estando presente en la litetura de Neurofisiología durante varios años. Rosenblatt, sin embargo, se aventuró en describir el modelo como un programa para computadoras y demostró con simples experimentos que dicho modelo era generalizable. El perceptrón fue construído como una solución a un problema particular dentro del aprendizaje automático, el reconocimiento de patrones. En el caso más sencillo, este problema consiste en hallar una regla para separar datos en dos categorías distintas a partir de ejemplos.

Para construir la regla de separación, el perceptrón sigue el modelo más sencillo de neurona, propuesto previamente por McCulloch y Pitts, de acuerdo al cual una neurona recibe n valores (o *inputs*) en la forma de un vector $x = (x^1, \ldots, x^n) \in X \subset \mathbb{R}^n$ y genera una etiqueta (*output*) $y \in \{-1, +1\}$ a través de una dependecia funcional dada por

$$y = sgn\{(w \cdot x) - b\}$$

con $\cdot$ el producto interno de vectores en $\mathbb{R}^n$, b un valor de límite y un vector

w que se genera en el proceso de aprendizaje. Geométricamente, una neurona divide el espacio X en dos regiones: en una la etiqueta y vale $+1$ y en la otra -1. Las dos regiones son separadas por el hiperplano:

$$(w \cdot x) - b = 0$$

El vector w y el escalar b determinan la posición del hiperplano y sus valores son aprendidos por el perceptrón. Cuando combinamos varias neuronas, el perceptrón separa el espacio de entrada en en dos regiones lineales a trozos y no necesariamente conexas. En 1960 no era claro como elegir todos los parametros $(w_1, \ldots, w_k)$ y $(b_1, \cdots, b_k)$ de todas las neuronas, por lo que se fijaban los valores de las primeras $k-1$ y se intentaba encontrar los valores deseables para la última de ellas. Geométricamente, se transformaba el espacio de entrada X en un nuevo espacio Z (eligiendo coeficientes apropiados para las primeras $k-1$ neuronas) y luego se utilizaban los datos de entrenamiento para construir un hiperplano que separe el plano Z. Tomando prestados de la fisiología los conceptos de aprendizaje con estímulos de premios y castigos, Rosenblat propuso un simple algoritmo para hallar estos coeficientes de manera iterativa, el cual describiremos a continuación.

DESCRIBIR perceptrón.

En 1962 Novikoff demostró el primer teorema relacionado al perceptrón. Podemos decir que este teorema inció propiamente la teoría del aprendizaje.

Teorema. *Dado un conjunto de datos de entrenamiento como el descrito previamente, de manera que*
1) La norma de los vectores de entrenamiento z_i está acotada por una constante R:

$$|z_i| \leq R, \qquad \forall i = 1, 2, \ldots, k$$

2) Los datos de entrenamiento pueden separados con un margen ρ:

$$\sup_{w} \min_{i} y_i (z_i \cdot w) > \rho$$

3) Los datos son alimentandos al perceptrón una cantidad suficiente de veces.

Entonces el algoritmo encuentra el hiperplano que separa los datos de entrenamiento, luego de a lo sumo N correcciones, con N verificando:

$$N \leq \left[\frac{R^2}{\rho^2} \right]$$

Resaltamos este teorema porque jugó un papel fundamental en la creación de la teoría del aprendizaje, conectando el principio de minimización de errores en el conjunto de datos de entrenamiento con la capacidad de generalización de los algoritmos de clasificación y su causa.

4. Hacia la formalización

Volviendo al caso de clasificación binaria en aprendizaje supervisado, partimos de ejemplos (datos de entrenamiento) en un espacio de entrada X con alguna de las dos posibles etiquetas del espacio $Y = \{-1, +1\}$. Aquí, *aprender* se reduce a estimar una relación funcional $f : X \to Y$, el clasificador. Un algoritmo de aprendizaje es aquel que a partir de los datos de entrenamiento construye una función f. Nos interesa construir una teoría que no asuma de manera estricta nada acerca de X e Y, pero nos permitimos asumir ciertas cosas del mecanismo que genera los datos de entrenamiento. En particular, asumiremos que existe una *distribución de probabilidad conjunta* $P = P(X, Y)$ sobre $X \times Y$ y que las muestras son tomadas de forma independiente de esta distribución de forma *iid* -independiente e idénticamente distribuída-. Notemos lo siguiente:

1. *No imponemos condiciones a la distribución de probabilidad P.* La gran diferencia que encontramos entre la Estadística tradicional y la teoría del aprendizaje estadístico es que en esta última trabajamos de manera agnóstica a la distribución que genera las muestras y deseamos llegar a conclusiones generales.

2. *Consideramos a las etiquetas de manera no determinística.* Consideramos a P como una distribución de probabilidad no solo sobre las instancias de X, si no también sobre las propias etiquetas de Y. Por lo tanto, estas últimas no son solo funciones tradicionales de los datos en X, si no que ellas mismas pueden ser aleatorias. Tenemos al menos dos buenas razones para tomar esta consideración: por un lado, el proceso de generación de datos puede tener ruido al asignar etiquetas (por ejemplo tomemos el caso de un detector de spam basado en la opinión de etiquetadores humanos que clasifican emails con un porcentaje de error; incluso los humanos pueden clasificar incorrectament algunos de esos emails), y por otro, ciertos problemas se prestan a que existan clases que se solapan (pensemos en la dificultad en diferenciar a un perro de un gato en una fotografías que los captura desde una gran distancia o con baja resolución).
   En la práctica, en vez de asignar etiquetas a los elementos en X de manera determinística, daremos la probabilidad condicional de la etiqueta y dado el valor x. En el caso de clasificación binaria, basta solo dar la probabilidad $P(Y = 1|X = x)$ de que la etiqueta tenga valor $Y = 1$, dado que la restante es complementaria:

   $$P(Y = -1|X = x) \quad = \quad 1 - P(Y = 1|X = x)$$

   Ciertos problemas que hagan uso de datos con etiquetas con poco ruido nos llevaran naturalmente a probabilidades condicionales cercanas a 0 y 1, dejando un margen de error pequeño, pero cuando tratemos con solapamiento de clases, las probabilidades condicionales pueden acercarse a $\frac{1}{2}$ para cada etiqueta. Independientemente de la causa, que las probabilidades condicionales sobre las etiquetas se acerquen a $\frac{1}{2}$ vuelve más dificil el aprendizaje, dado que crece el número de errores del clasificador.

3. *Muestro independiente.* Una de las condiciones más fuertes que imponemos en la teoría del aprendizaje estadístico es que asumimos que las muestras son tomadas de forma independiente. En muchas aplicaciones, esta suposición está justificada, pero hay ramas muy importantes de la disciplina en donde esto no se cumple, por ejemplo en el análisis de series de tiempo, en donde la secuencialidad de los datos viola la condición de iid (cada valor depende en alguna medida de los anteriores). Esto es también cierto para las aplicaciones a lenguaje natural, y constituye una de las razones principales por las cuales esta rama más moderna del aprendizaje tiene bases teóricas menos fuertes que el aprendizaje automático tradicional.

4. *La distribución P es fija.* Al no considerar al tiempo como un parámetro, ni existir un orden en las muestras, asumimos que la distribución que las origina es siempre la misma. Esto, como en el punto anterior, no se cumple en las series de tiempo. Otro caso donde se viola esta suposición es en aquellos problemas en donde la distribución de probabilidad de los datos de entrenamiento no coincide con el de los datos posteriores, llamado *covariate shift*, por ejemplo en un sistema de *scoring* de usuarios de una empresa que crece súbitamente y que suma a personas que no se corresponden a los perfiles que existían originalmente en su base de datos (e.g. se admite que inmigrantes no bancarizados y sobre los que no hay datos previos accedan a préstamos).

5. *La distribución P es desconocida al momento de aprender.* Si conociéramos de antemano la probabilidad condicional P, el problema del aprendizaje sería trivial pues podríamos siempre determinar el mejor clasificador posible (aunque no sea perfecto, dada la naturaleza aleatoria de las etiquetas). Solo tenemos acceso a P de manera indirecta, a través de las muestras. Intuitivamente, esto nos hace pensar que, consiguiendo un número lo suficientemente grande de muestras, podemos aproximar las propiedades de la distribución P, pero con errores. Uno de los principales logros de la teoría del aprendizaje estadístico es brindarnos un marco teórico para acotar este error.

4.1. Pérdida y Riesgo

Para saber qué tan bien se comporta un clasificador f, necesitamos medir sus equivocaciones. Para esto, definiremos una *función de pérdida*, ℓ, que le asigne un valor al hecho de que f clasifique a cierto $x \in X$ con la etiqueta $y \in Y$. Llamermos *costo* a dicho valor, dado que más adelante penalizaremos al clasificador en base a los errores que cometa a través del alogritmo de aprendizaje.
El ejemplo más sencillo de función de pérdida es la "pérdida-0-1", que le asigna un costo de 0 a una instancia de clasificación correcta y un costo 1 a una incorrecta, es decir:

$$\ell(X, Y, f(X)) := \begin{cases} 1 & \text{si } f(X) \neq Y \\ 0 & \text{si } f(X) = Y \end{cases}$$

En problemas de regresión, la función de pérdida más conocida es el *error cuadrático*, dado por

$$\ell(X, Y, f(X)) := (Y - f(X))^2$$

Por convención, una pérdida igual a 0 implica una clasificación perfecta y valores mayores implican peor clasificación. Es decir, el aprendizaje suele implicar un desafío de optimización en dónde deseamos hallar el mínimo de la función de perdida.

Mientras que la función de pérdida mide el error del clasificador en un punto individual $x \in X$, llamamos *riesgo*, $\mathcal{R}$, del clasificador a la pérdida esperada sobre todos los datos generados por la distribución de probabilidad P. Es decir

$$\mathcal{R}(f) := E\left(\ell(X, Y, f(X))\right)$$

Desde luego que otra función g es un mejor clasificador que f para un problema dado si su riesgo es más bajo, es decir si $\mathcal{R}(g) < \mathcal{R}(f)$, por lo que el mejor clasificador de todos es aquel con el riesgo más bajo.

Algo que no hemos considerado aún es si los clasificadores f tienen alguna característica especial. Para formalizarlo, tomaremos funciones de un espacio de funciones $\mathcal{F}$ que aplican X en Y. En un principio, parecería aceptable tomar el espacio de todas las funciones posibles, o más precisamente, el conjunto de todas las funciones *medibles* que aplican X en Y, $\mathcal{F}_{all} = \{f \text{ medibles} \mid f : X \to Y\}$. En este caso, podemos señalar cuál es el clasificador ideal, dada la distribución P, al que llamamos *clasificador Bayesiano*, f_{Bayes}, y al que definimos como:

$$f_{\text{Bayes}} := \begin{cases} 1 & \text{si } P(Y = 1|X = x) \geq \frac{1}{2} \\ -1 & \text{en otro caso} \end{cases}$$

Observermos que, en caso de que las etiquetas fueran determinísticas, es decir donde $P(Y = y|X = x) = 1$ para cada $x \in X$ con su respectiva etiqueta y, f_{Bayes} eligiría correctamente en todos los casos. De existir un ligero solapamiento de clases de tal manera que para un cierto x tengamos $P(Y = 1|X = x) = 0{,}9$, entonces tendríamos que en la mayoría de los casos la etiqueta de x es $+1$ y, siendo este el valor elegido por f_{Bayes}, el clasificador Bayesiano estaría en lo correcto.

En la práctica no es posible computar directamente el clasificador Bayesiano dado que, como dijimos, la distribución de probabilidad conjunta P es desconocida. Sin embargo, este clasificador es una herramienta teórica que nos permite formular el problema estandar de la clasificación binaria, el cual es:

Dado un conjunto de datos de entrenamiento $\{(X_1, Y_1), \ldots, (X_n, Y_m)\}$ *obtenidos iid de una distribución* P*, y dada una función de pérdida* ℓ*, deseamos construir una funión clasificadora* $f : X \to Y$ *cuyo riesgo* $\mathcal{R}(f)$ *sea lo más cercano posible al riesgo de* f_{Bayes}*.*

Notemos que no solo es imposible computar el error del clasificador Bayesiano, sino también el propio riesgo de cualquier clasificador f. Es decir, dado un problema definido (minimizar el riesgo del clasificador), con una solución ideal que podemos escribir (el propio clasificador Bayesiano), no tenemos manera de computar ninguna cosa de utilidad. Aquí es donde la teoría de aprendizaje estadístico nos permite llegar a resultados y obtener garantías de la utilidad de esas soluciones.

4.2. Generalización

Dado que no conocemos la distribución de probabilidad $P(X, Y)$, no podemos calcular la esperanza de la pérdida de un clasificador cualquiera f, es decir su riesgo. Lo que si podemos hacer, dado un conjunto de entrenamiento, es "contar"(o medir, en general, para problemas que no son de clasificación binaria) el número de errores del clasificador sobre los datos de entrenamiento. A esta cantidad le daremos el nombre de *riesgo empírico* y lo veremos presente en la literatura también como *error de entrenamiento*. Lo definimos como

$$R_{emp}(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, f(X_i))$$

Por ejemplo, de la función de pérdida llamada error cuadrático se deriva el riesgo empírico *error cuadrático medio*, que es ampliamente utilizado en la práctica.

Usualmente, un algoritmo de aprendizaje aceptable es capaz de producir un clasificador f que performa aceptablemente bien en un conjunto de datos conocido, es decir tal que el riesgo empírico del clasificador es bajo. Nos interesa que un clasificador f tenga riesgo bajo en todo el espacio de entrada X, no solo en los datos de entrenamiento. Decimos que un clasificador *generaliza* bien si la diferencia entre su riesgo y su riesgo empírico es baja.

DEFINICIÓN Generalización

Por supuesto que una buena generalización no asegura que el riesgo, ni el riesgo empírico, sean bajos, si no que dichas cantidades son cercanas. Pero lo que nos interesa es que el riesgo empírico sea un buen estimativo del riesgo del clasificador, y esto es lo que nos permitirá hacer afirmaciones acerca del error del clasificador en la práctica.

4.3. Consistencia

Intuitivamente, parece razonable pedirle a un algoritmo de aprendizaje que, al ser presentado con más y más ejemplos, converja a una solución óptima. En Estadística, la noción de *consistencia* se relaciona con la capacidad de hacer afirmaciones con respecto a lo que sucede en el límite de una cantidad infinita de muestras y, a diferencia de la generalización, que es acerca de una función

en particular, es una propiedad de un conjunto de funciones. Para ilustrar este concepto, denotemos como f_n al clasificador construido por un algoritmo de aprendizaje luego de ser presentado con n puntos de entrenamiento. Por el momento no repararemos en cómo el algoritmo construye esta función, pero podemos estar seguro que la elige de un cierto espacio funcional $\mathcal{F}$. Más adelante veremos que dicho espacio puede estar explícitamente dado, como en el caso de la regresión lineal, o no, y existir implícitamente en el mecanismo del propio algoritmo, como es el caso de las redes neuronales. Más allá de si $\mathcal{F}$ es explícito o no, el algoritmo debe elegir a la mejor función en dicho espacio basándose en los puntos de entrenamiento. Por otro lado, sabemos precisamente cuál es en teoría el mejor clasificador en $\mathcal{F}$: el que tiene el menor riesgo. Por simplicidad, supongamos que este es único (no tiene por qué serlo, puede no haber un solo mínimo global para la función de riesgo). Definimos a este clasificador óptimo como:

$$f_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} \mathcal{R}(f)$$

Considerando que el clasificador bayesiano introducido en la sección anterior es el mejor clasificador posible, podríamos denotarlo, siguiendo la misma convención, como $f_{\mathcal{F}_{all}}$. Desde luego que el espacio $\mathcal{F}$ que elegimos puede no contenerlo, por lo que $\mathcal{R}(f_{\text{Bayes}}) < \mathcal{R}(f_{\mathcal{F}})$. Con estas ideas, podemos definir los distintos de convergencia que trataremos al considerar el concepto de consistencia.

Definición 1. *Sea $(X_i, Y_i)_{i \in \mathbb{N}}$ una sucesión infinita de puntos de entrenamiento que han sido tomados iid de una distribución P. Sea ℓ una función de pérdida. Por cada $n \in \mathbb{N}$, sea f_n un clasificador construído por algún algoritmo de aprendizaje usando los primeros n puntos de entrenamiento. Sea $\mathcal{F}$ el espacio funcional de todos los posibles clasificadores según el mecanismo de construcción del algoritmo. Entonces*

1. *El algoritmo de aprendizaje se dice* consistente con respecto a $\mathcal{F}$ y a P *si el riesgo $\mathcal{R}(f_n)$ converge en probabilidad al riesgo $\mathcal{R}(f_{\mathcal{F}})$ del mejor clasificador en $\mathcal{F}$. Esto es, que para todo $\epsilon > 0$:*

$$P\left(\mathcal{R}(f_n) - \mathcal{R}(f_{\mathcal{F}}) > \epsilon\right) \to 0 \qquad \textit{cuando} \qquad n \to \infty$$

2. *El algoritmo de aprendizaje se dice* Bayes-consistente con respecto a P *si el riesgo $(R)(f_n)$ converge en probabilidad al riesgo $(R)(f_{Bayes})$ del clasificador bayesiano. Esto es, para todo $\epsilon > 0$*

$$P\left(\mathcal{R}(f_n) - \mathcal{R}(f_{Bayes}) > \epsilon\right) \to 0 \qquad \textit{cuando} \qquad n \to \infty$$

3. *El algoritmo de aprendizaje se dice* universalmente consistente con respecto a $\mathcal{F}$ *si es consistente con respecto a $\mathcal{F}$ para toda distribución de probabilidad P.*

4. *El algoritmo de aprendizaje se dice* universalmente Bayes-consistente *si es Bayes-consistente para toda distribución de probabilidad P.*

4.4. Overfitting y Underfitting

5. Bibliografía

[1] *Statistical Learning Theory: Models, Concepts and Results* - von Luxburg, Schölkopf (2008)

[2] *The Nature of Statistical Learning Theory, second edition* - Vladimir Vapnik (2000)