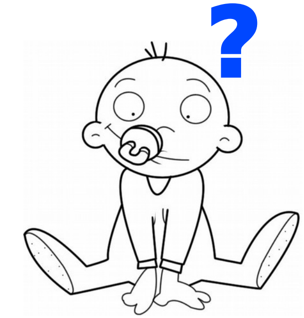
Aprendizaje Automático
Segundo Cuatrimestre de 2016

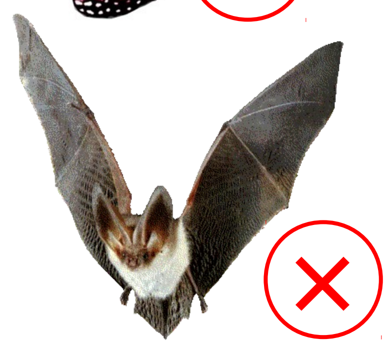
Aprendizaje de Conceptos

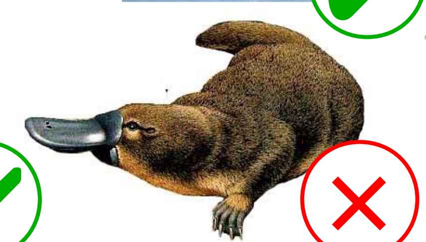
DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Aprendiendo un Concepto…



- ¿Qué es un “ave”?

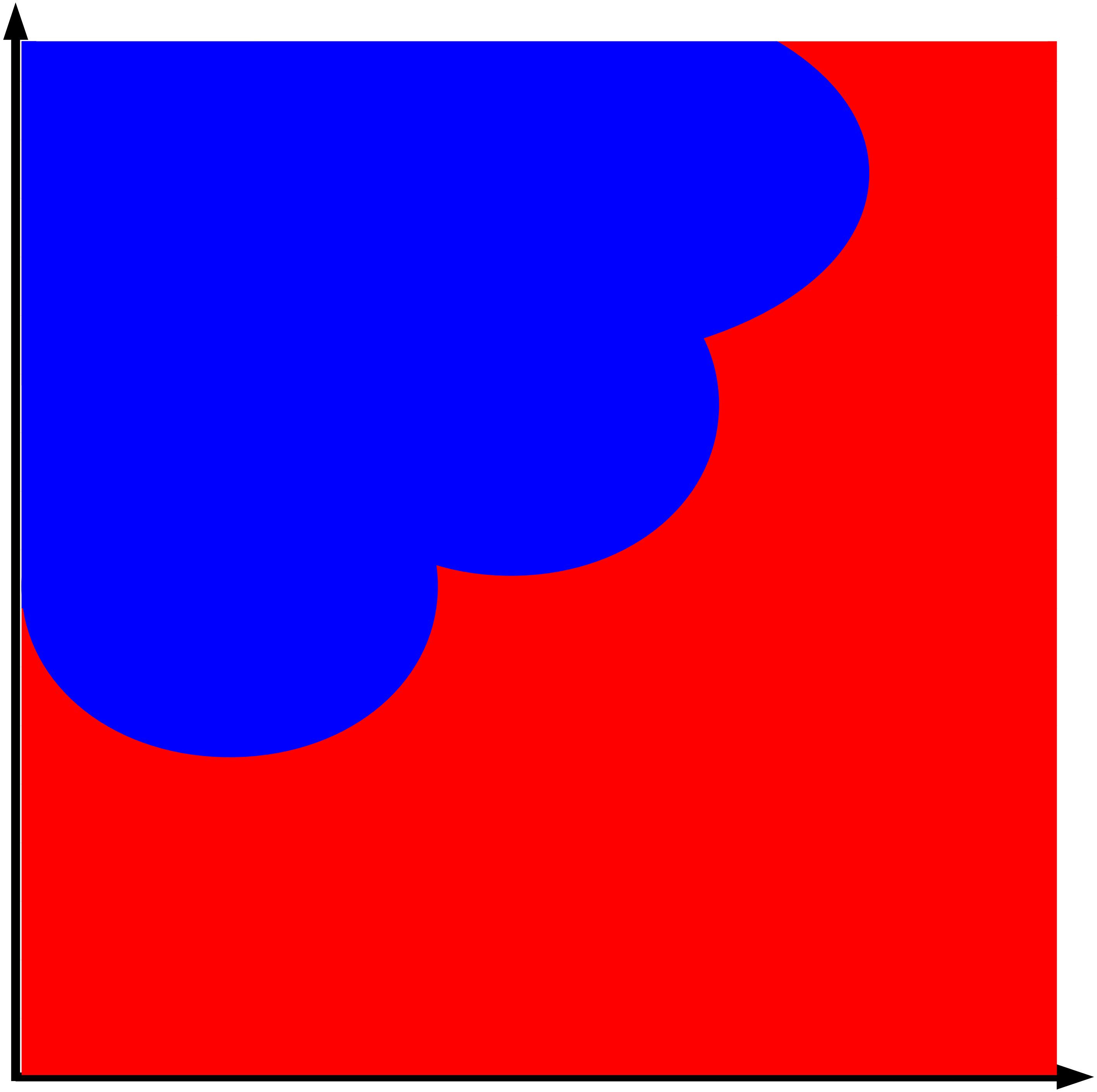




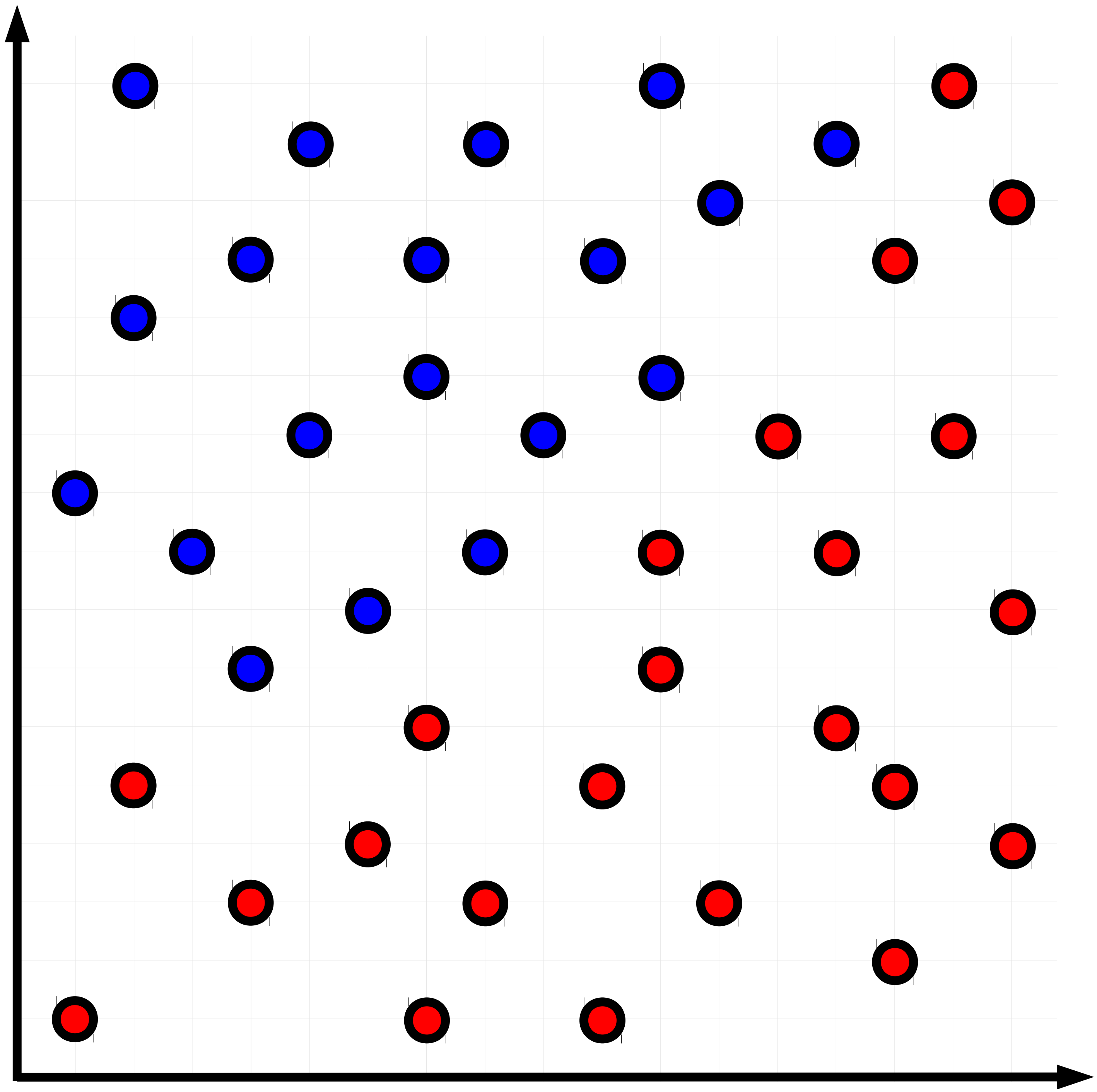
Aprendizaje de Conceptos

- Aprender un concepto: inducir una función booleana a partir de ejemplos de entrenamiento.
- Construimos, evaluamos y refinamos hipótesis:
  - h_{Ave} = Vuela
  - h'_{Ave} = (Vuela ∨ DosPatas) ∧ TienePico
  - h''_{Ave} = DosPatas ∧ TienePlumas ∧ PoneHuevos
  - ...
- La forma que tienen estas hipótesis define el espacio de hipótesis *H*.
- Puede ocurrir que *H* no contenga al concepto objetivo.
- Algoritmo de aprendizaje: buscar la hipótesis en *H* que mejor se ajuste a los datos de entrenamiento.

$$\boxed{\underset{h \in H}{\operatorname{argmax}} \; P(h \mid D)}$$



- Este es el **concepto** *(función objetivo)* que queremos aprender.
- **<u>Es desconocido!!</u>**
- Sólo podemos conocerlo a través de muestras *(instancias)*.



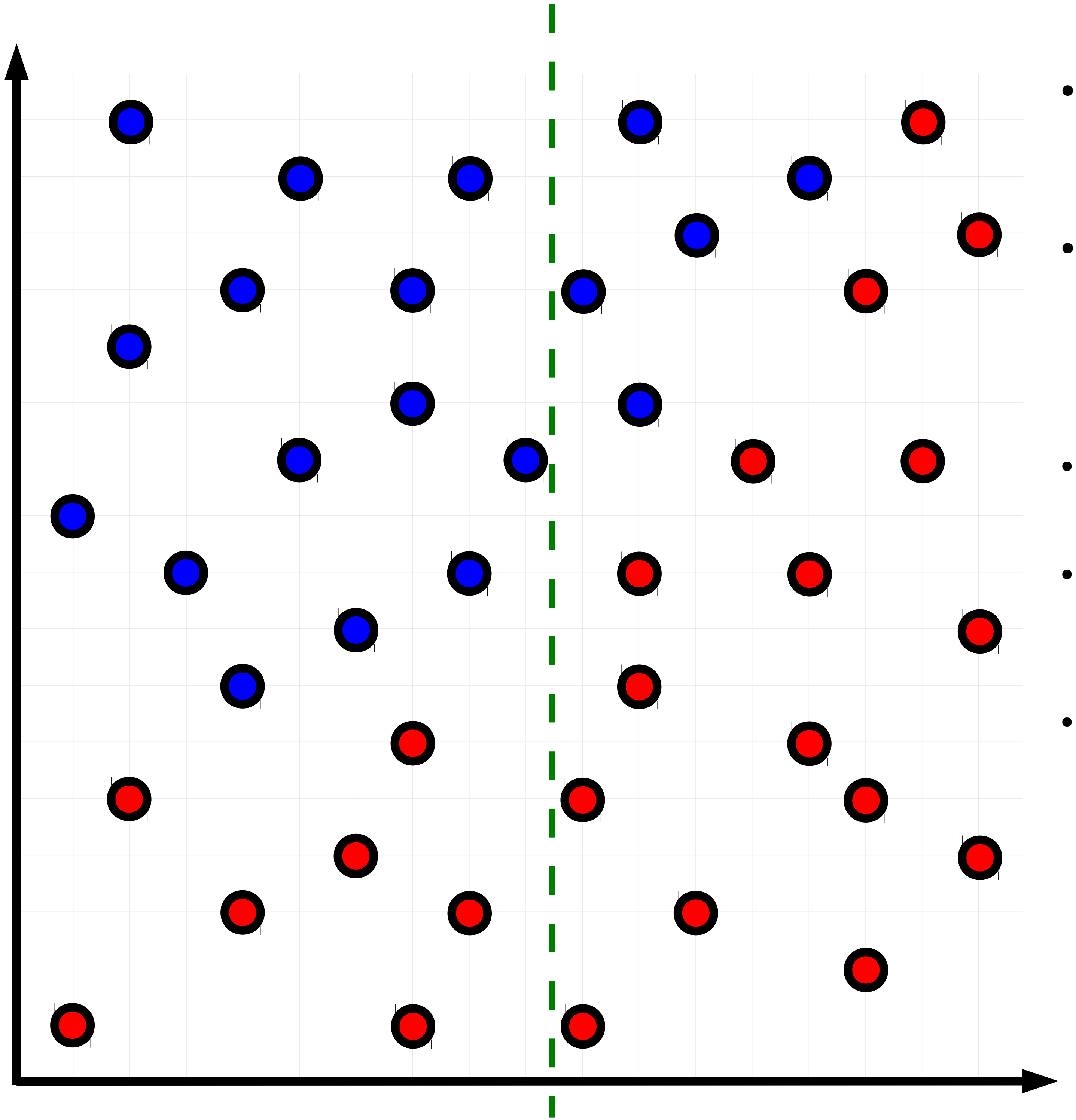
- Tomamos N muestras *(instancias)*.
- Cada punto tiene dos coordenadas: (x, y) *(atributos)*.
- Cada punto tiene un color: **azul** o **rojo** *(clases)*.
- Buscamos una función $h(x, y) \rightarrow$ color *(hipótesis)* que aproxime a la función objetivo.

Aprendizaje Inductivo

- Consiste en construir un modelo general a partir de información específica.

- **Hipótesis de Aprendizaje Inductivo**

  Cualquier hipótesis que aproxime bien a una función objetivo sobre un conjunto suficientemente grande de instancias de entrenamiento también aproximará bien a la función objetivo sobre instancias no observadas.



- Buscamos:
  Función $h(x, y) \rightarrow$ color

- <u>Primera idea:</u>

  $h(x, y)$ =

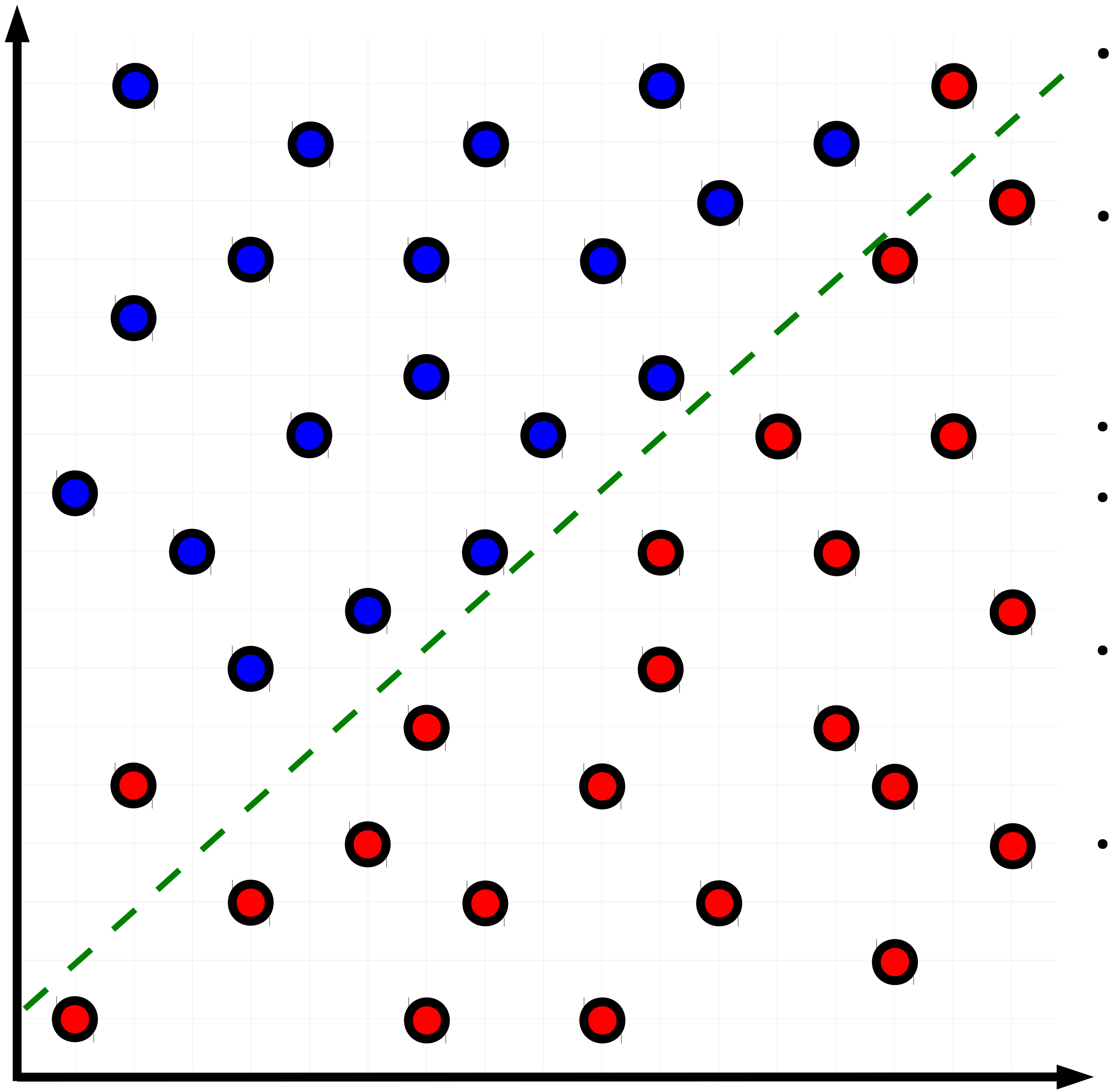
  azul si $x < k$

  rojo en caso contrario

- H = posibles valores de
  < *k*, *color_izq* >

- Algoritmo: buscar
  < *k*, *color_izq* > que mejor separe las 2 clases.

- Obs: El concepto objetivo no tiene esta forma.



- Buscamos:
  Función $h(x, y) \rightarrow$ color

- <u>Segunda idea:</u>

  $h(x, y) =$

  azul si $y > m\,x + b$

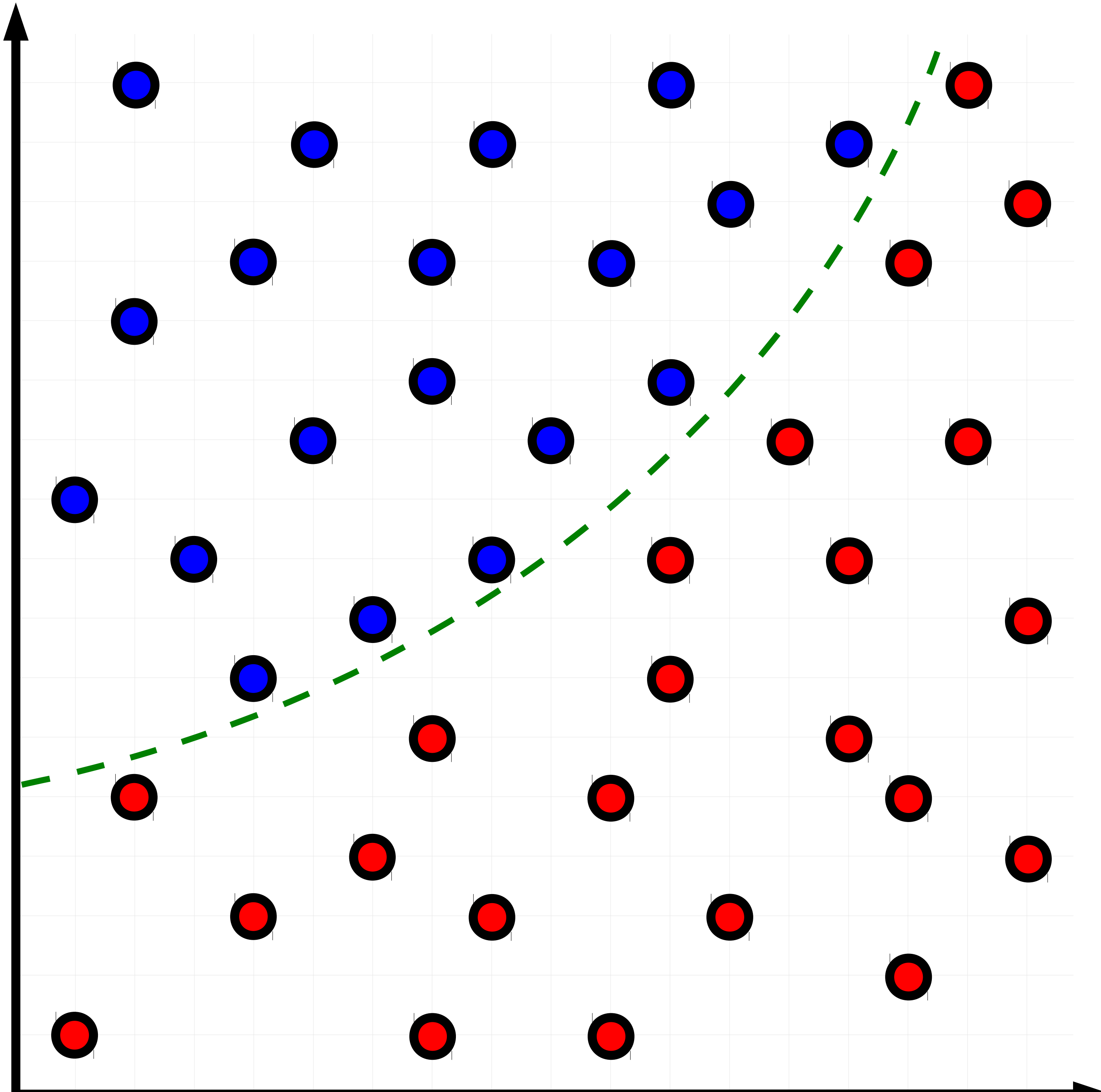
  rojo en caso contrario

- H = < *m*, *b*, *color_sup* >

- Algoritmo: buscar
  < *m*, *b*, *color_sup* > que mejor separe las 2 clases.

- El concepto objetivo tampoco tiene esta forma, pero la aproximación parece mejor que la anterior.

- Encontrar los valores *m*, *b* es más costoso.



- Buscamos:
  Función $h(x, y) \rightarrow$ color

- <u>Tercera idea:</u>

  $h(x, y)$ =
  azul si $y > a\, x^2 + b\, x + c$
  rojo en caso contrario

- H = < *a*, *b*, *c*, *color_sup* >

- Algoritmo: buscar
  < *a*, *b*, *c*, *color_sup* > que
  mejor separe las 2 clases.

- El concepto objetivo tampoco tiene esta forma.

- La aproximación parece mejor que las otras dos.

- Encontrar los valores *a*, *b*, *c* es aún más costoso.

Sesgo Inductivo

- Los datos de entrenamiento no alcanzan para *inferir* un modelo. Hay infinitas posibilidades.
- Formalmente:

  L: algoritmo de aprendizaje automático

  X: conjunto de instancias

  c: concepto a aprender, definido sobre *X*

  $D = \{\langle x, c(x)\rangle\}$ conjunto de instancias de entrenamiento

  $L(x, D)$: clasificación asignada a *x* por *L* (entrenado sobre *D*)

$$(D \wedge x) \not\vdash L(x, D)$$

Sesgo Inductivo

- El sesgo inductivo de un algoritmo de aprendizaje es el conjunto de afirmaciones que el algoritmo utiliza para clasificar instancias nuevas.
- Formalmente:

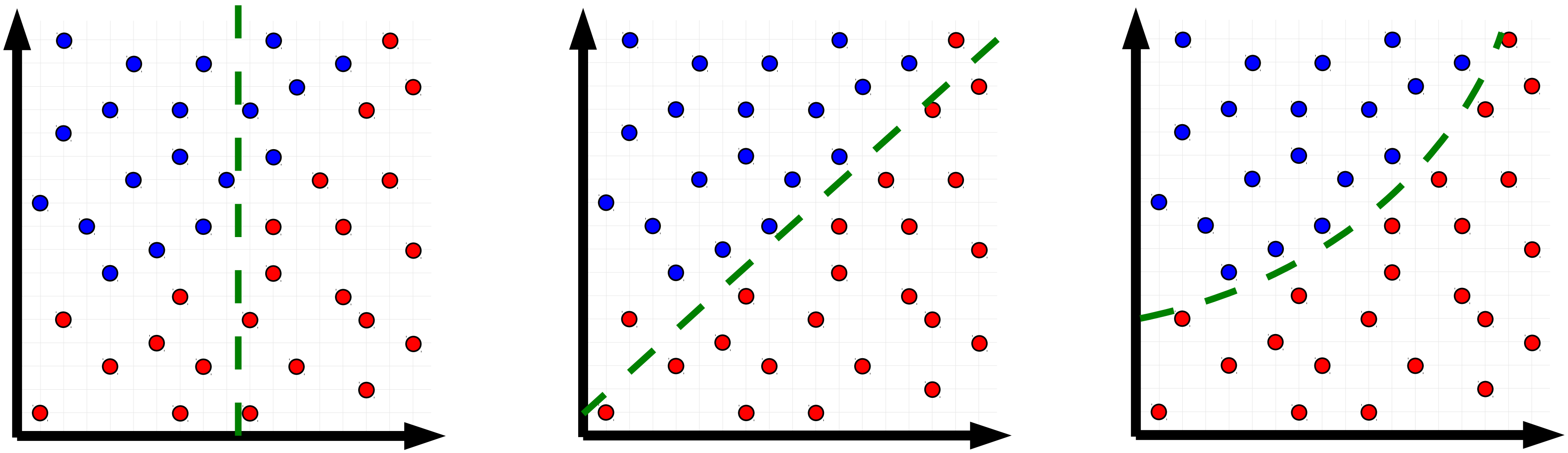
  L: algoritmo de aprendizaje automático
  X: conjunto de instancias
  c: concepto a aprender, definido sobre *X*
  D = {⟨*x*, *c*(*x*)⟩} conjunto de instancias de entrenamiento
  L(*x*, *D*): clasificación asignada a *x* por *L* (entrenado sobre *D*)
  B: conjunto minimal de afirmaciones tal que

$$(\forall x \in X)\ (B \wedge D \wedge x)\ \vdash\ L(x, D)$$

- *B* incluye la forma de las hipótesis y características de funcionamiento del algoritmo.

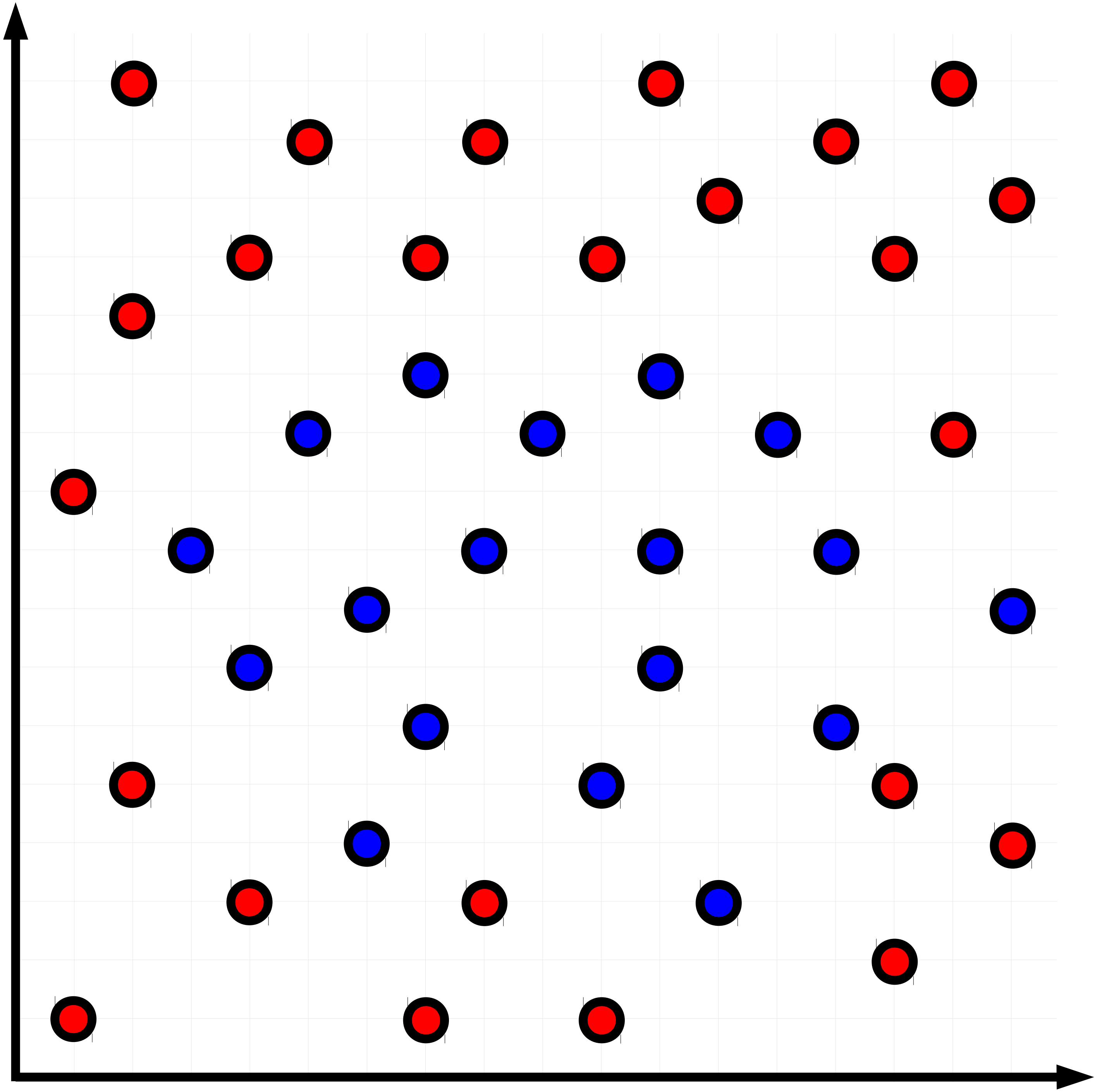
$$(\forall x \in X)\ ({\color{red}B} \land D \land x) \vdash\ L(x, D)$$

- En cada ejemplo, el sesgo inductivo *B* incluye que las hipótesis *(funciones discriminantes)* son…

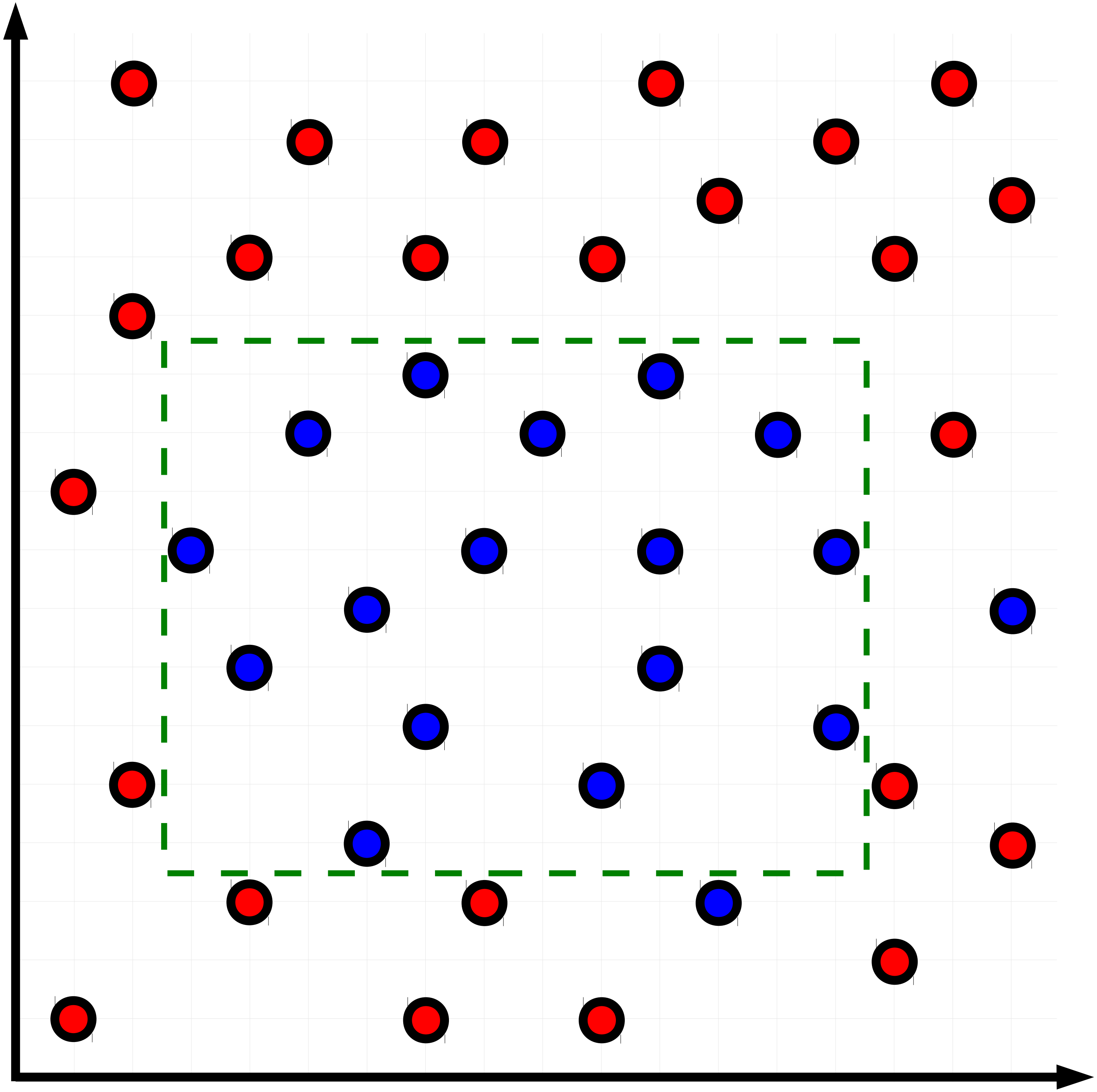


Rectas verticales — Rectas — Parábolas

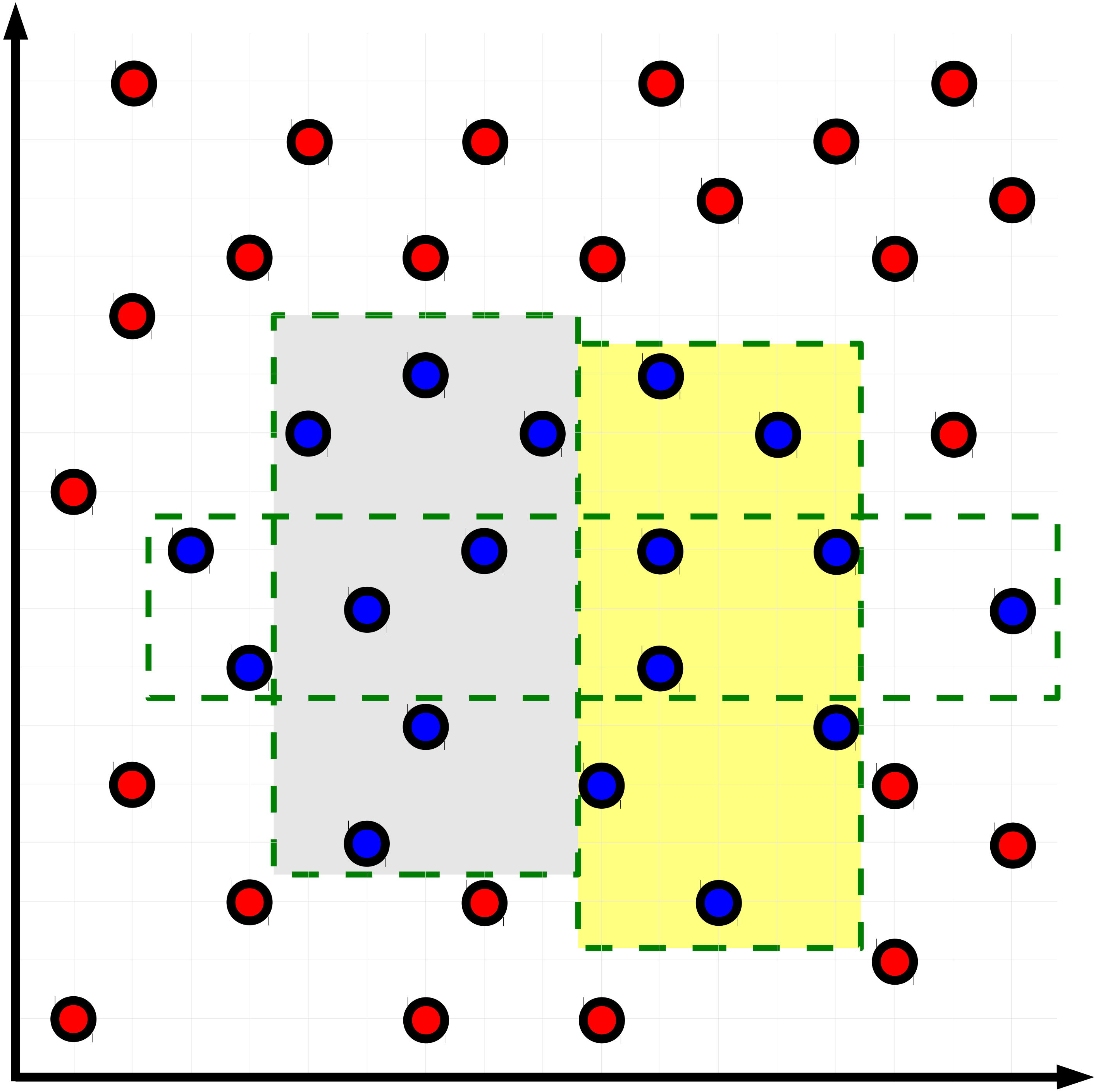
- Además, *B* debe incluir la manera en que cada algoritmo desempata entre hipótesis equivalentes.
  - Ej: poner la línea tan a la derecha como sea posible.



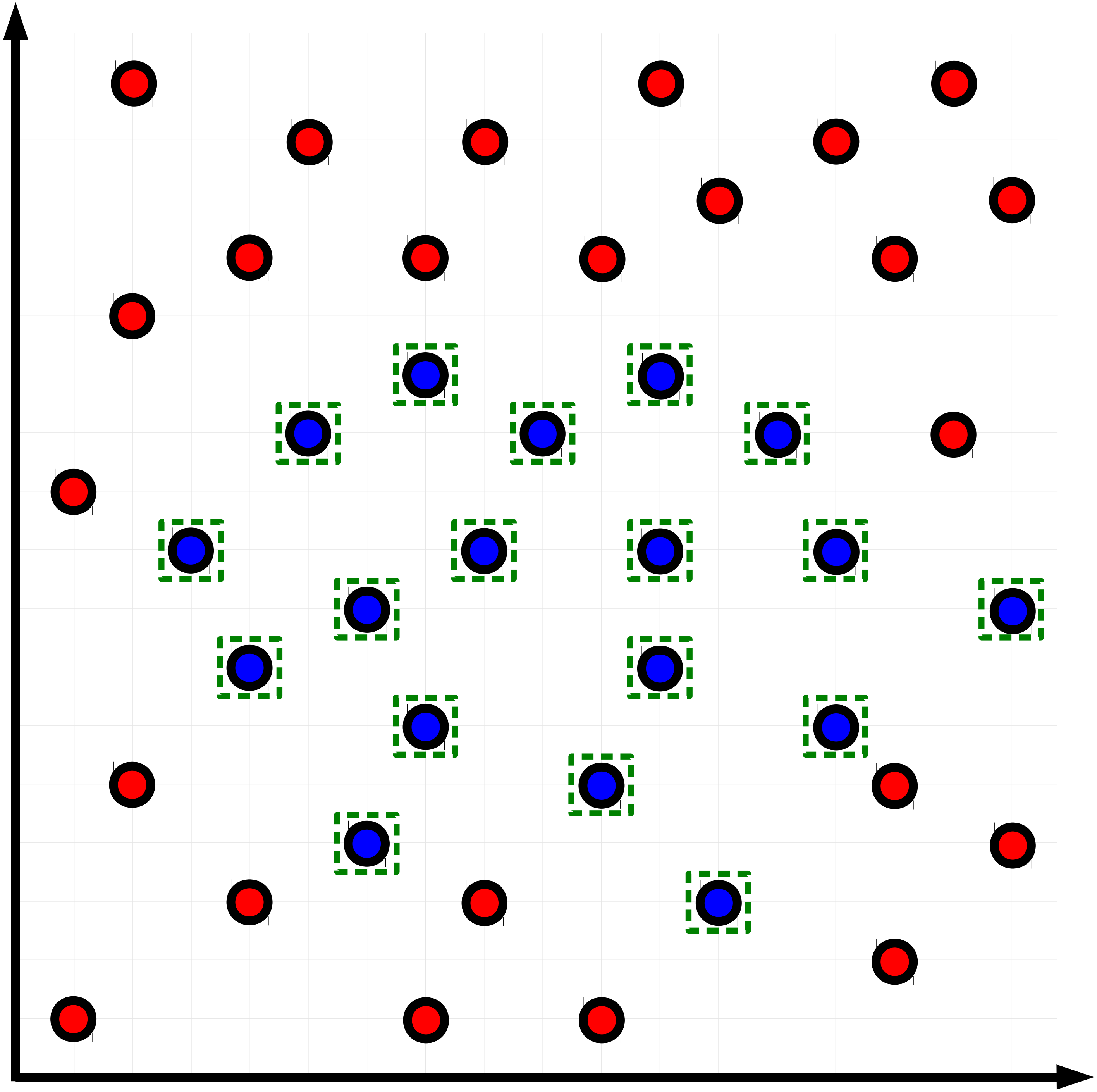
- **Nuevo ejemplo.**
- Queremos aprender un **concepto** *(función objetivo)* desconocido.
- Tenemos un conjunto de instancias de entrenamiento.



- Elegimos un algoritmo cuyas hipótesis tienen forma rectangular.
- Cada rectángulo tiene:
  - Base (*b*)
  - Altura (*h*)
  - Posición (*x*, *y*)
  - Color interior (azul/rojo)
- Espacio de hipótesis
  - H = < *b*, *h*, *x*, *y*, *color* >



- Ahora, elegimos otro algoritmo con hipótesis formadas por <u>*varios*</u> rectángulos.



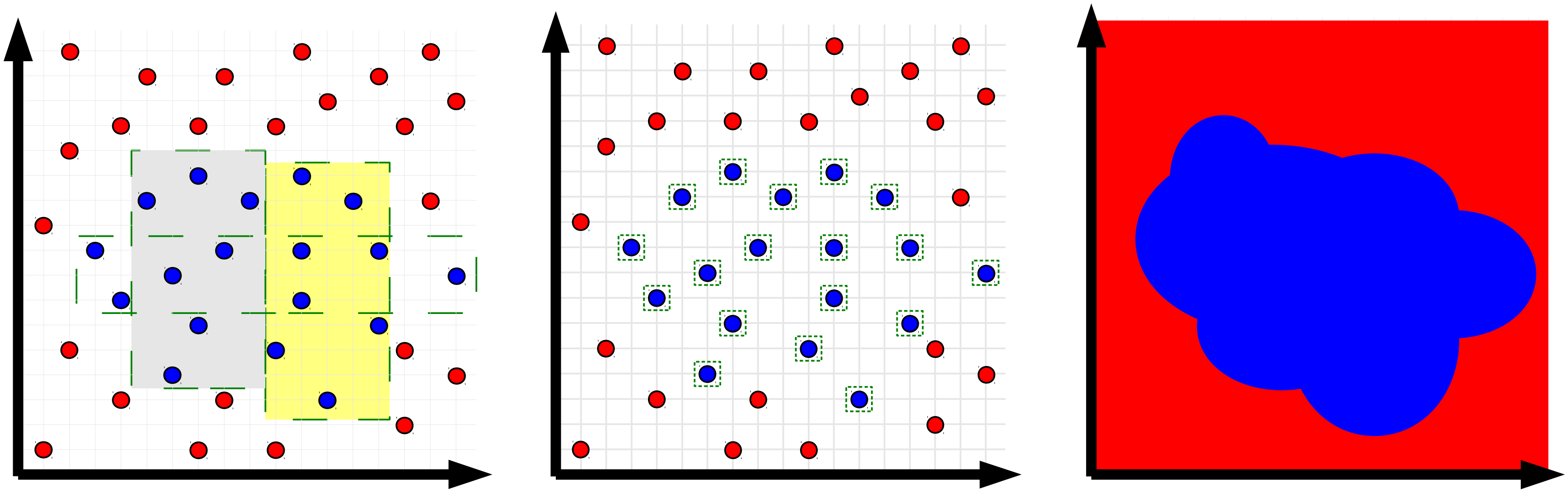
- Ahora, elegimos otro algoritmo con hipótesis formadas por *varios* rectángulos.
- Riesgo de sobreajuste (*overfitting*).

Navaja de Occam
(o de Ockham)

- *“Pluralitas non est ponenda sine necessitate.”*
  - La pluralidad no debe postularse sin necesidad.
- En igualdad de condiciones, elegir la explicación más simple.
- Es un principio metodológico (no una ley!).
- Aprendizaje Automático:
  - Ante dos hipótesis que se ajustan igualmente bien a los datos de entrenamiento, es esperable que la más simple generalice mejor.

Navaja de Occam
(o de Ockham)

- Aprendizaje Automático:
  - Ante dos hipótesis que se ajusten igualmente bien a los datos de entrenamiento, es esperable que la más simple generalice mejor.



Fragmentos de “Funes el Memorioso”

Jorge Luis Borges, Ficciones, 1944

“Nosotros, de un vistazo, percibimos tres copas en una mesa; Funes, todos los vástagos y racimos y frutos que comprende una parra.”

“Funes no sólo recordaba cada hoja de cada árbol de cada monte, sino cada una de las veces que la había percibido o imaginado.”

“Era casi incapaz de ideas generales, platónicas. No sólo le costaba comprender que el símbolo genérico 'perro' abarcara tantos individuos dispares de diversos tamaños y diversa forma; le molestaba que el perro de las tres y catorce (visto de perfil) tuviera el mismo nombre que el perro de las tres y cuarto (visto de frente).”

“Había aprendido sin esfuerzo el inglés, el francés, el portugués, el latín. Sospecho, sin embargo, que no era muy capaz de pensar. Pensar es olvidar diferencias, es generalizar, abstraer. En el abarrotado mundo de Funes no había sino detalles, casi inmediatos.”

Repaso de Probabilidades

$P(A)$ Probabilidad de ocurrencia del evento A

$P(\neg A) = 1 - P(A)$

$P(A \wedge B) = P(A \cap B) = P(A, B)$

$P(A \mid B)$ Probabilidad de que ocurra A suponiendo que B ocurre

$$P(A \wedge B) = P(A \mid B) \cdot P(B) = P(B \mid A) \cdot P(A)$$

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Teorema de Bayes

Si $A_1, A_2, \ldots, A_n$ es una partición del espacio muestral, entonces:

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i) \cdot P(A_i)$$

Teorema de Probabilidad Total
(Marginalización)

Aprendizaje Automático

Verosimilitud de D

Probabilidad
a priori

$$\underbrace{h_{\mathrm{MAP}}}_{\text{Máxima a posteriori}} = \underset{h \in H}{\operatorname{argmax}}\ \underbrace{P(h \mid D)}_{\text{Probabilidad a posteriori}} = \underset{h \in H}{\operatorname{argmax}}\ \frac{P(D \mid h) \cdot P(h)}{\cancel{P(D)}}$$

Máxima
a posteriori

Probabilidad
a posteriori

P(D) es constante
respecto de *h*

Si suponemos $P(h) = \frac{1}{|H|}\ \forall h \in H$, entonces:

$$h_{\mathrm{ML}} = \underset{h \in H}{\operatorname{argmax}}\ P(D \mid h)$$

Máxima
verosimilitud

Resumen

- Aprendizaje de conceptos.
- Función objetivo; instancias; clases; atributos.
- Hipótesis; espacio de hipótesis.
- Aprendizaje inductivo.
- Hipótesis de aprendizaje inductivo.
- Sesgo inductivo.
- Sobreajuste; Navaja de Occam.
- Hipótesis MAP, ML.
- Probabilidad a priori, a posteriori, verosimilitud.