Selección de Modelo

Fernando Lozano

Universidad de los Andes

22 de septiembre de 2017

Minimización de error y complejidad

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta:

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$$

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$$

- Intuición: Hipótesis $h \in \mathcal{H}$ que minimiza error en los datos sobre suficientes datos, produce un error $e(h)$ pequeño.

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$$

- Intuición: Hipótesis $h \in \mathcal{H}$ que minimiza error en los datos sobre suficientes datos, produce un error $e(h)$ pequeño.
- Queremos balancear la complejidad de $\mathcal{H}$ con el ajuste de $h \in \mathcal{H}$ a los datos de entrenamiento:

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$$

- Intuición: Hipótesis $h \in \mathcal{H}$ que minimiza error en los datos sobre suficientes datos, produce un error $e(h)$ pequeño.
- Queremos balancear la complejidad de $\mathcal{H}$ con el ajuste de $h \in \mathcal{H}$ a los datos de entrenamiento:
    - $\mathcal{H}$ muy simple puede no contener una buena aproximación a la función que queremos aprender

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$$

- Intuición: Hipótesis $h \in \mathcal{H}$ que minimiza error en los datos sobre suficientes datos, produce un error $e(h)$ pequeño.
- Queremos balancear la complejidad de $\mathcal{H}$ con el ajuste de $h \in \mathcal{H}$ a los datos de entrenamiento:
  - $\mathcal{H}$ muy simple puede no contener una buena aproximación a la función que queremos aprender
  - $\mathcal{H}$ muy compleja puede ajustarse bien a los datos pero predecir pobremente.

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$$

- Intuición: Hipótesis $h \in \mathcal{H}$ que minimiza error en los datos sobre suficientes datos, produce un error $e(h)$ pequeño.
- Queremos balancear la complejidad de $\mathcal{H}$ con el ajuste de $h \in \mathcal{H}$ a los datos de entrenamiento:
  - $\mathcal{H}$ muy simple puede no contener una buena aproximación a la función que queremos aprender
  - $\mathcal{H}$ muy compleja puede ajustarse bien a los datos pero predecir pobremente.
- Crítico cuando:

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}} [h(x) \neq y]$$

- Intuición: Hipótesis $h \in \mathcal{H}$ que minimiza error en los datos sobre suficientes datos, produce un error $e(h)$ pequeño.
- Queremos balancear la complejidad de $\mathcal{H}$ con el ajuste de $h \in \mathcal{H}$ a los datos de entrenamiento:
  - $\mathcal{H}$ muy simple puede no contener una buena aproximación a la función que queremos aprender
  - $\mathcal{H}$ muy compleja puede ajustarse bien a los datos pero predecir pobremente.
- Crítico cuando:
  - Número de datos es pequeño.

Minimización de error y complejidad

- Datos $(x, y) \sim \mathcal{D}$
- clase de hipótesis $\mathcal{H}$ (p.ej. Redes Neuronales con arquitectura dada).
- Meta: encontrar $h \in \mathcal{H}$ que minimiza error

$$e(h) = \mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$$

- Intuición: Hipótesis $h \in \mathcal{H}$ que minimiza error en los datos sobre suficientes datos, produce un error $e(h)$ pequeño.
- Queremos balancear la complejidad de $\mathcal{H}$ con el ajuste de $h \in \mathcal{H}$ a los datos de entrenamiento:
  - $\mathcal{H}$ muy simple puede no contener una buena aproximación a la función que queremos aprender
  - $\mathcal{H}$ muy compleja puede ajustarse bien a los datos pero predecir pobremente.
- Crítico cuando:
  - Número de datos es pequeño.
  - Datos ruidosos.

Algoritmos de selección de modelo

- Complejidad de la clase de modelos es una variable a determinar por el algoritmo de aprendizaje.

Algoritmos de selección de modelo

- Complejidad de la clase de modelos es una variable a determinar por el algoritmo de aprendizaje.
- Considere la secuencia anidada de clases de hipótesis:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

Algoritmos de selección de modelo

- Complejidad de la clase de modelos es una variable a determinar por el algoritmo de aprendizaje.
- Considere la secuencia anidada de clases de hipótesis:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

- Selección de modelo procede en dos pasos:

Algoritmos de selección de modelo

- Complejidad de la clase de modelos es una variable a determinar por el algoritmo de aprendizaje.
- Considere la secuencia anidada de clases de hipótesis:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

- Selección de modelo procede en dos pasos:
    1. Seleccione una función candidata h_i de cada clase $\mathcal{H}_i$ (usualmente minimizando criterio de error empírico en $\mathcal{H}$).

Algoritmos de selección de modelo

- Complejidad de la clase de modelos es una variable a determinar por el algoritmo de aprendizaje.
- Considere la secuencia anidada de clases de hipótesis:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

- Selección de modelo procede en dos pasos:
  1. Seleccione una función candidata h_i de cada clase $\mathcal{H}_i$ (usualmente minimizando criterio de error empírico en $\mathcal{H}$).
  2. Use algún criterio para seleccionar $h \in \{h_1, h_2, \ldots, h_d, \ldots\}$ tal que $e(h)$ sea pequeño.

Validación cruzada

Validación cruzada

- Estimación directa de $e(h_i)$

Validación cruzada

- Estimación directa de $e(h_i)$
  1. Datos S ise dividen en subconjuntos S_{train} and S_{test}, con $|S_{train}| = (1-\gamma)|S|$ y $|S_{test}| = \gamma|S|$, $\gamma \in (0,1)$.

Validación cruzada

- Estimación directa de $e(h_i)$
  1. Datos S ise dividen en subconjuntos S_{train} and S_{test}, con $|S_{train}| = (1-\gamma)|S|$ y $|S_{test}| = \gamma|S|$, $\gamma \in (0,1)$.
  2. Se halla hipótesis candidata en $h_d \in \mathcal{H}_d$ minimizando error empírico (o función sustituta) enS_{train}.

Validación cruzada

- Estimación directa de $e(h_i)$
  1. Datos S ise dividen en subconjuntos S_{train} and S_{test}, con $|S_{train}| = (1-\gamma)|S|$ y $|S_{test}| = \gamma|S|$, $\gamma \in (0,1)$.
  2. Se halla hipótesis candidata en $h_d \in \mathcal{H}_d$ minimizando error empírico (o función sustituta) enS_{train}.
  3. Se selecciona la hipótesis candidata h_d con el menor error empírico en S_{test}:

$$h_{d^\star} = \arg\min_{\{h_1,h,2,\ldots,\}} \hat{e}_{S_{test}}(h_d)$$

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq$$

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2}\ln\frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?
    - Muy pequeño $\Rightarrow$

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?
  - Muy pequeño $\Rightarrow$ estimación pobre de $e(h)$.

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?
  - Muy pequeño $\Rightarrow$ estimación pobre de $e(h)$.
  - Muy grande $\Rightarrow$

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?
  - Muy pequeño $\Rightarrow$ estimación pobre de $e(h)$.
  - Muy grande $\Rightarrow$ aprendizaje pobre.

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?
  - Muy pequeño $\Rightarrow$ estimación pobre de $e(h)$.
  - Muy grande $\Rightarrow$ aprendizaje pobre.
  - Típicamente $\gamma \approx 0{,}1$.

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?
    - Muy pequeño $\Rightarrow$ estimación pobre de $e(h)$.
    - Muy grande $\Rightarrow$ aprendizaje pobre.
    - Típicamente $\gamma \approx 0{,}1$.
- Estimativo de $e(h_d)$ es usualmente ruidoso.

- Usando cotas de Chernoff, sabemos que para estimar $e(h)$ con precisión ε y confianza $1-\delta$ requerimos

$$|S_{test}| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

- En la práctica es posible que no tengamos suficientes datos.
- Selección de γ?
    - Muy pequeño $\Rightarrow$ estimación pobre de $e(h)$.
    - Muy grande $\Rightarrow$ aprendizaje pobre.
    - Típicamente $\gamma \approx 0{,}1$.
- Estimativo de $e(h_d)$ es usualmente ruidoso.
- En la práctica se usa validación cruzada k-múltiple.

Validación Cruzada k-múltiple

- Idea es suavizar estimativo de $e(h)$

Validación Cruzada k-múltiple

- Idea es suavizar estimativo de $e(h)$
- Para una clase $\mathcal{H}$:

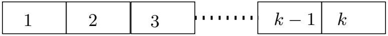
Validación Cruzada k-múltiple

- Idea es suavizar estimativo de $e(h)$
- Para una clase $\mathcal{H}$:
  1. S se divide en $S_1, S_2, \ldots, S_k$.

1	2	3	....	$k-1$	k

Validación Cruzada k-múltiple

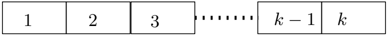
- Idea es suavizar estimativo de $e(h)$
- Para una clase $\mathcal{H}$:
  1. S se divide en $S_1, S_2, \ldots, S_k$.



  2. Para cada $i = 1, 2, \ldots, k$

Validación Cruzada k-múltiple

- Idea es suavizar estimativo de $e(h)$
- Para una clase $\mathcal{H}$:
  1. S se divide en $S_1, S_2, \ldots, S_k$.



  2. Para cada $i = 1, 2, \ldots, k$
     1. Se halla h_i minimizando error empírico en $\bigcup_{j \neq i} S_j$

Validación Cruzada k-múltiple

- Idea es suavizar estimativo de $e(h)$
- Para una clase $\mathcal{H}$:
  1. S se divide en $S_1, S_2, \ldots, S_k$.

1	2	3	....	$k-1$	k

  2. Para cada $i = 1, 2, \ldots, k$
     1. Se halla h_i minimizando error empírico en $\bigcup_{j \neq i} S_j$
     2. Se estima error calculando error empírico $\hat{e}_{S_i}(h_i)$

Validación Cruzada k-múltiple

- Idea es suavizar estimativo de $e(h)$
- Para una clase $\mathcal{H}$:
  1. S se divide en $S_1, S_2, \ldots, S_k$.

1	2	3	....	$k-1$	k

  2. Para cada $i = 1, 2, \ldots, k$
     1. Se halla h_i minimizando error empírico en $\bigcup_{j \neq i} S_j$
     2. Se estima error calculando error empírico $\hat{e}_{S_i}(h_i)$
     3. Se promedian valores obtenidos:

$$\hat{e}(h_d) = \frac{1}{k} \sum_{i=1}^{k} \hat{e}_{S_i}(h_i)$$

Validación Cruzada k-múltiple

- Idea es suavizar estimativo de $e(h)$
- Para una clase $\mathcal{H}$:
  1. S se divide en $S_1, S_2, \ldots, S_k$.

1	2	3	········	$k-1$	k

  2. Para cada $i = 1, 2, \ldots, k$
     1. Se halla h_i minimizando error empírico en $\bigcup_{j \neq i} S_j$
     2. Se estima error calculando error empírico $\hat{e}_{S_i}(h_i)$
     3. Se promedian valores obtenidos:

$$\hat{e}(h_d) = \frac{1}{k} \sum_{i=1}^{k} \hat{e}_{S_i}(h_i)$$

- Para $d^\star$ que corresponde al menor valor de $\hat{e}(h_d)$, se halla h minimizando error empírico en $\mathcal{H}_{d^\star}$ en S.

Validación cruzada k-múltiple

Validación cruzada k-múltiple

- Procedimiento es costoso computacionalmente.

Validación cruzada k-múltiple

- Procedimiento es costoso computacionalmente.
- Usado ampliamente en la práctica.

Validación cruzada k-múltiple

- Procedimiento es costoso computacionalmente.
- Usado ampliamente en la práctica.
- Carece de soporte teórico, es un problema abierto importante.

Validación cruzada k-múltiple

- Procedimiento es costoso computacionalmente.
- Usado ampliamente en la práctica.
- Carece de soporte teórico, es un problema abierto importante.
- Errores no son v.a. normales, no son independientes.

Minimización de Riesgo Estructurado (SRM)

Minimización de Riesgo Estructurado (SRM)

- Secuencia anidada de clases de hipótesis:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

Minimización de Riesgo Estructurado (SRM)

- Secuencia anidada de clases de hipótesis:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

- Función candidata h_d de cada clase $\mathcal{H}_d$ minimiza error empírico en $\mathcal{H}_d$)

Minimización de Riesgo Estructurado (SRM)

- Secuencia anidada de clases de hipótesis:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

- Función candidata h_d de cada clase $\mathcal{H}_d$ minimiza error empírico en $\mathcal{H}_d$)
- Se escoge d^* de acuerdo a:

$$d^* = \arg\min_d \hat{e}(h_d) + p(d)$$

Minimización de Riesgo Estructurado (SRM)

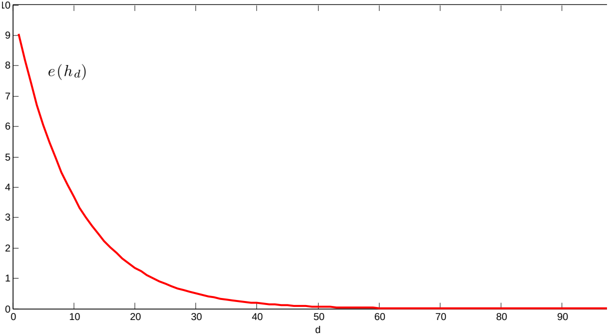
- Secuencia anidada de clases de hipótesis:

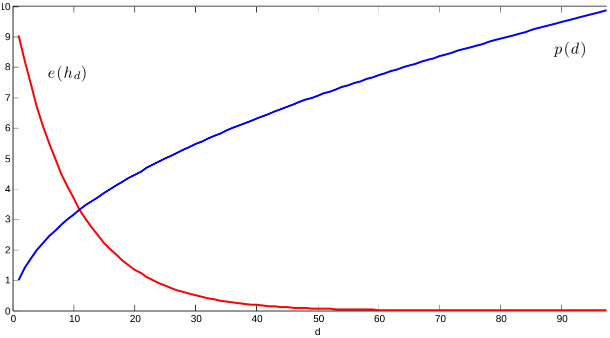
$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \mathcal{H}_d \subseteq \cdots$$

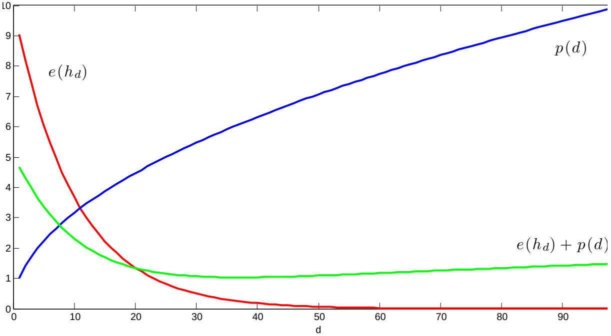
- Función candidata h_d de cada clase $\mathcal{H}_d$ minimiza error empírico en $\mathcal{H}_d$)
- Se escoge d^* de acuerdo a:

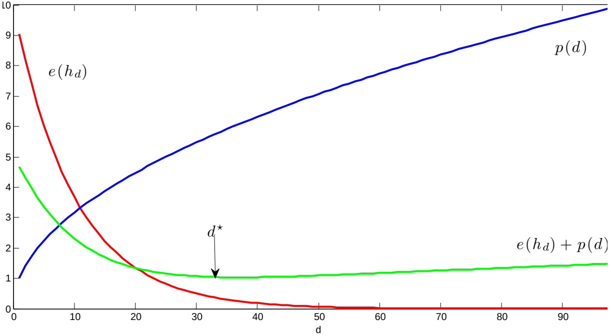
$$d^* = \arg\min_d \hat{e}(h_d) + p(d)$$

donde $p(d)$ es una función creciente de d que penaliza funciones de complejidad alta.


10
9
8
7
6
5
4
3
2
1
0
0
10
20
30
40
50
60
70
80
90
d
$e(h_d)$


$e(h_d)$
$p(d)$
d
0
10
20
30
40
50
60
70
80
90
1
2
3
4
5
6
7
8
9


$e(h_d)$
$p(d)$
$e(h_d) + p(d)$
10
9
8
7
6
5
4
3
2
1
0
0
10
20
30
40
50
60
70
80
90
d


$e(h_d)$
$p(d)$
$d^\star$
$e(h_d) + p(d)$
0
1
2
3
4
5
6
7
8
9
0 10 20 30 40 50 60 70 80 90
d

Usando la dimensión VC

- Si $VC(\mathcal{H}_d) = d$ con alta probabilidad, tenemos

Usando la dimensión VC

- Si $VC(\mathcal{H}_d) = d$ con alta probabilidad, tenemos

$$\sup_{h \in \mathcal{H}_d} |e(h) - \hat{e}(h))| \leq O\left(\sqrt{\frac{d \log m}{m}}\right)$$

Usando la dimensión VC

- Si $VC(\mathcal{H}_d) = d$ con alta probabilidad, tenemos

$$\sup_{h \in \mathcal{H}_d} |e(h) - \hat{e}(h))| \leq O\left(\sqrt{\frac{d \log m}{m}}\right)$$

- Podemos escoger $p(d) = O\left(\sqrt{\frac{d \log m}{m}}\right)$

Usando la dimensión VC

- Si $VC(\mathcal{H}_d) = d$ con alta probabilidad, tenemos

$$\sup_{h \in \mathcal{H}_d} |e(h) - \hat{e}(h))| \leq O\left(\sqrt{\frac{d \log m}{m}}\right)$$

- Podemos escoger $p(d) = O\left(\sqrt{\frac{d \log m}{m}}\right)$
- Más precisamente, la complejidad óptima se escoge de acuerdo a la regla:

$$d^* = \arg\min_d \left\{ \hat{e}(d) + \frac{d(\frac{\ln(2m)}{d} + 1)}{m}(1 + \sqrt{(1 + \frac{\hat{e}(d)m}{d(\frac{\ln(2m)}{d} + 1)})} \right\}$$

SRM usando la dimensión VC

SRM usando la dimensión VC

- $p(d)$ no depende de $\mathcal{D}$.

SRM usando la dimensión VC

- $p(d)$ no depende de $\mathcal{D}$.
- Es la misma penalización para cualquier distribución de los datos.

SRM usando la dimensión VC

- $p(d)$ no depende de $\mathcal{D}$.
- Es la misma penalización para cualquier distribución de los datos.
- En casos prácticos no se conoce la dimensión VC sino sólo una cota superior.

SRM usando la dimensión VC

- $p(d)$ no depende de $\mathcal{D}$.
- Es la misma penalización para cualquier distribución de los datos.
- En casos prácticos no se conoce la dimensión VC sino sólo una cota superior.
- Constantes no son óptimas.

SRM usando la dimensión VC

- $p(d)$ no depende de $\mathcal{D}$.
- Es la misma penalización para cualquier distribución de los datos.
- En casos prácticos no se conoce la dimensión VC sino sólo una cota superior.
- Constantes no son óptimas.
- En la práctica es dificil balancear error y penalización.

SRM usando la dimensión VC

- $p(d)$ no depende de $\mathcal{D}$.
- Es la misma penalización para cualquier distribución de los datos.
- En casos prácticos no se conoce la dimensión VC sino sólo una cota superior.
- Constantes no son óptimas.
- En la práctica es dificil balancear error y penalización.
- Tiende a sobre penalizar.