Tecnologías para Inteligencia Artificial (247101009)
Tema 1. Regresión lineal

Javier Vales Alonso
Máster Universitario en Ingeniería Telemática

2020

Universidad Politécnica de Cartagena

Tabla de contenido

¿Cómo estudiar esta unidad?

1. Haga una primera lectura de las diapositivas de la unidad. Concéntrese en ver las ideas generales y hacer una primera revisión de las matemáticas.
2. Haga una revisión a fondo de las matemáticas con las diapositivas y resuelva en el notebook los ejercicios indicados. Intente comprender todos los desarrollos involucrados. En caso de dudas, lea el referencias sugeridas (ver referencias en Tema 0) o contacte al profesor.
3. Finalmente, envie el notebook a través de AV.

Introducción

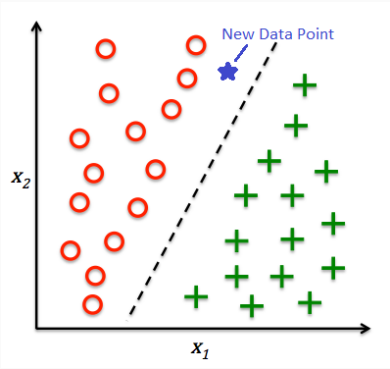
Introduction

Los problemas de aprendizaje supervisados se caracterizan por la disponibilidad de un *conjunto de datos etiquetados* $\{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \ldots, (\boldsymbol{x}_N, t_N)\}$, donde las instancias $\{\boldsymbol{x}_n\}$, para $n = 1, \ldots, N$ son variables D-dimensionales para las cuales se conoce la salida (o *target*) correspondiente $\{t_n\}$.

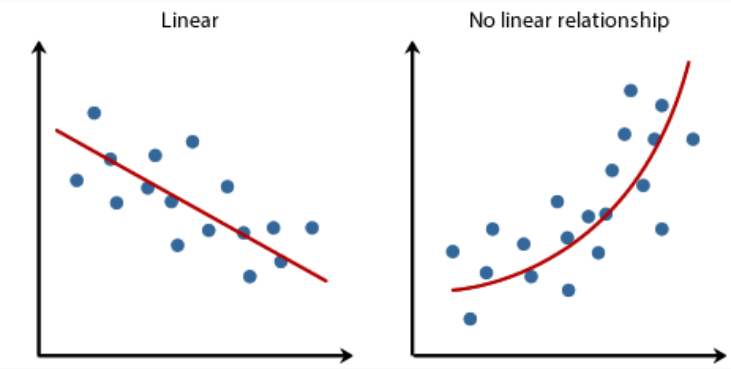
El objetivo de los métodos de aprendizaje supervisado es proporcionar predicciones de t para nuevas instancias $\boldsymbol{x}$.

Si t toma valores en un conjunto finito de valores (por ejemplo, SPAM/NO SPAM, ALEGRÍA/TRISTEZA/SORPRESA, etc.) el problema de aprendizaje se llama *clasificación*, que puede ser binario o multiclase. Cuando t toma valores en un dominio continuo, el problema se llama *regresión*.

Introducción (II)



Ejemplo de clasificación binaria.
La salida prevista es una clase (círculo rojo o cruz verde).



Ejemplo de regresión lineal.
La salida pronosticada es un número real (la línea o curva rojas).

Introducción (III)

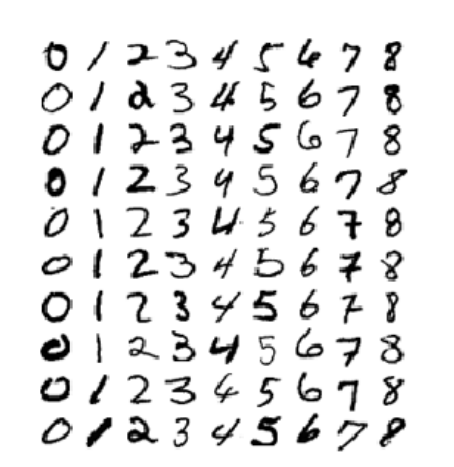
Frecuentemente, en problemas de clasificación, el etiquetado se realiza manualmente. Esto puede ser costoso en términos de tiempo y recursos. Por ejemplo, un médico especialista examinando imágenes MRI para identificar posibles patologías.

En otros casos, el *target* se conoce *a posteriori* y luego es agregado al conjunto de datos, e.g., en un problema de pronóstico diario en bolsa los precios reales de las acciones se conocen al día siguiente.

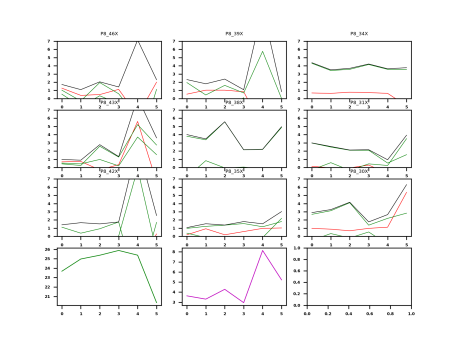
Otra posibilidad es obtener los targets utilizando un proceso independiente a la obtención de los datos de entrada. Por ejemplo, el paralaje estelar se utiliza para obtener rangos de distancia a las estrellas cercanas. Estos valores pueden emplearse como targets en un problema de aprendizaje que trate de predecir esas distancias a partir de otras variables.

Introducción (IV)

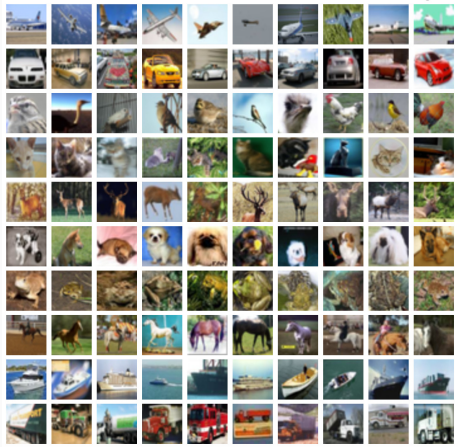
Etiquetado asociado a algunos de los conjuntos de datos de ejemplo introducidos en la Unidad 0:



Cada instancia en la base de datos MNIST está etiquetada con el dígito representado en la imagen, de 0 a 9.



En el conjunto de datos de saco de boxeo, cada instancia está etiquetada con el tipo de golpe correspondiente: JAB/CROSS/HOOK



En el conjunto de datos CIFAR-10, cada imagen está etiquetada con la clase (una de los siguientes 10: AVIÓN/AUTOMÓVIL/PÁJARO/GATO/CIERVO/PERRO/-RANA/CABALLO/BARCO/-CAMIÓN)

Algortimos de aprendizaje supervisado

Los principales algoritmos de aprendizaje supervisado son:

1. **Regresión lineal**. Módelo paramétrico básico para problemas de regresión. En nuestra exposición partiremos de ideas intuitivas, para mostrar después cómo este modelo se deriva de una base estadística. Además, se introducen conceptos centrales de ML, como el overfitting y el underfitting, la regularización, el cross-validation, o el balance del sesgo/varianza.
2. **Regresión logística**. Módelo paramétrico básico para problemas de clasificación con frontera lineal. Lo estudiaremos en el Tema 2.

Algortimos de aprendizaje supervisado (II)

3. *K*-**Nearest Neighbors**. Método basado en instancias (i.e., no paramétrico) se describe en el Tema 3.
4. **Árboles de decisión y bosques aleatorios**. Los árbol de decisión son un modelo de caja blanca (es fácil para una persona entender el proceso de toma de decisiones). Los bosques aleatorios son conjuntos de árboles de decisión. Ambos se presentan también en el Tema 3.

Algortimos de aprendizaje supervisado (III)

7. **Máquinas de vector soporte**. Este método permite la creación de clasificadores lineales de margen máximo. Se estudiarán en el Tema 4.
8. **Redes neuronales artificiales**. Este método permite la creación de clasificadores y regresores no lineales. Se estudiarán en el Tema 5.

Método de ajuste de curvas

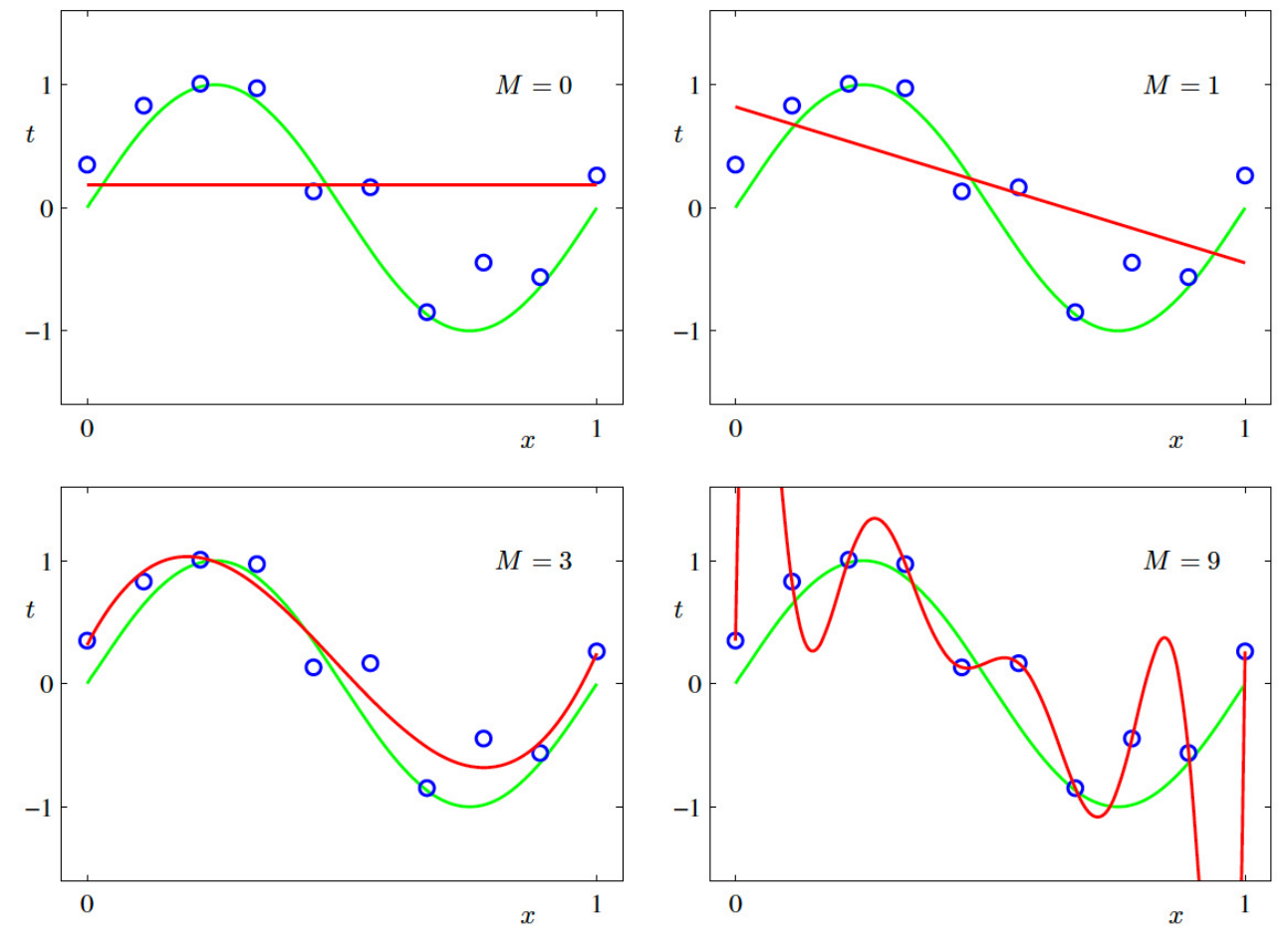
Método de ajuste de curvas

La forma más simple (y algo limitada) de estudiar problemas de regresión es considerarlos como problemas *de ajuste de curvas*. Esto es, seleccionar de una familia de funciones candidatas la curva que predice mejor los targets observados.

Surgen varias preguntas en este punto:

- ¿Cómo se pueden definir las funciones candidato o *hipótesis*?
- ¿Cómo medir el error entre la hipótesis y los targets?
- ¿Hasta qué punto podemos confiar en las predicciones?

Método de ajuste de curvas (II)



Ejemplos de ajuste de curvas. El conjunto de entrenamiento de $N = 10$ puntos que se muestra como círculos azules es unidimensional en x. Los targets son $t = sin(2\pi x)$ (la curva verde) más un pequeño ruido aleatorio gaussiano. Los ajustesj corresponden a bases de funciones polinómicas de grado M (curvas rojas). **Tómese un momento para considerar las preguntas anteriores en vista de esta figura.**

Funciones de hipótesis

Una manera simple de construir funciones de hipótesis es a partir de una base de funciones $\{\phi_j\}$, para $j = 1, \ldots, M-1$, y asociar a cada una de ellas un *peso* w_j. De esta manera la hipótesis[1] se escribe como:

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\boldsymbol{x}) \tag{1}$$

Donde w_0 se llama el parámetro de *bias*, y $\boldsymbol{w}$ denota el vector $(w_0, \ldots, w_{M-1})^T$. Al asumir una función de base ficticia $\phi_0(\boldsymbol{x}) = 1$, la última ecuación se puede reescribir simplemente como:

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\boldsymbol{x}) \tag{2}$$

[1] Incluso si las funciones de la base no son lineales en $\boldsymbol{x}$ nuestro enfoque de regresión sigue siendo llamado *lineal* ya que $y(\boldsymbol{x}, \boldsymbol{w})$ es lineal en w ya que $y(\boldsymbol{x}, \lambda_1 \boldsymbol{w}_1 + \lambda_2 \boldsymbol{w}_2) = \lambda_1 y(\boldsymbol{x}, \boldsymbol{w}_1) + \lambda_2 y(\boldsymbol{x}, \boldsymbol{w}_2)$.

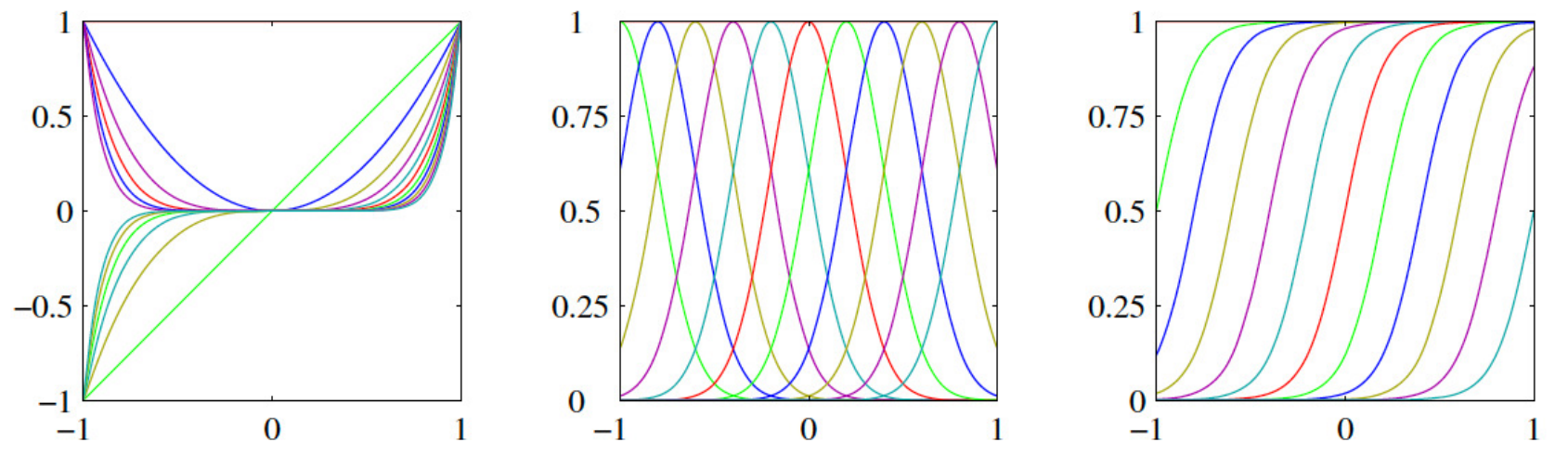
Funciones de hipótesis (II)

La base de funciones se pueden seleccionar entre muchas opciones posibles. Por ejemplo, lo más trivial es configurar $\phi_j(\boldsymbol{x}) = (\boldsymbol{x})_j$, i.e., sólo el valor de la característica j^{th}, y en este caso, al ser $\boldsymbol{x}$ D-dimensional es $M - 1 = D$.

Otra opción podría ser dejar que la base de funciones sean polinomios de grado K (por tanto, $M - 1 = K^D$)

Otras opciones comunes son: Funciones gaussianas, Funciones sigmoidales, Base de Fourier (funciones sinusoidales localizadas en el tiempo/espacio), o Wavelets (funciones localizadas tanto en tiempo/espacio como en frecuencia).

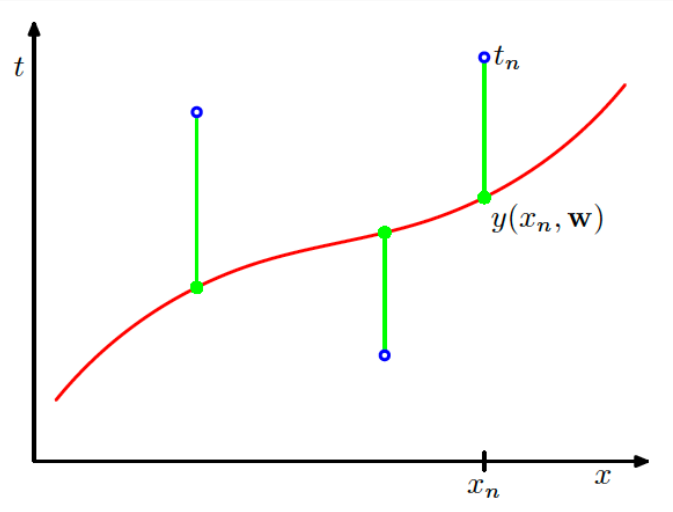
Hipótesis funciones (III)



Ejemplos de base de funciones polinómica (izquierda), gaussiana (centro), y sigmoidales (derecha)

Función de coste

La forma "intuitiva" para medir el error (*loss*) en el ajuste de la curva es usando el error cuadrático (SE) entre los targets t_n y las predicciones $y(\boldsymbol{x}_n, \boldsymbol{w})$ en el conjunto de entrenamiento.



$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}[y(\boldsymbol{x}_n, \boldsymbol{w}) - t_n]^2 = \frac{1}{2}\sum_{n=1}^{N}[\sum_{j=0}^{M-1} w_j \phi_j(\boldsymbol{x}_n) - t_n]^2 \quad (3)$$

donde se agrega el factor $\frac{1}{2}$ por conveniencia posterior y no afecta a los pesos óptimos.

Minimización de costes

Sea $\boldsymbol{\phi}(\boldsymbol{x})$ el vector $(\phi_0(\boldsymbol{x}), \ldots, \phi_{M-1}(\boldsymbol{x}))^T$. Entonces,

$$J(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n) - t_n)^2 \tag{4}$$

Los pesos óptimos $\boldsymbol{w}^*$ (correspondientes al MSE mínimo) se pueden calcular estableciendo el gradiente $\nabla J(\boldsymbol{w}) = 0$:

$$\nabla J(\boldsymbol{w}) = \sum_{n=1}^{N} (\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n) - t_n) \boldsymbol{\phi}(\boldsymbol{x}_n)^T = 0 \tag{5}$$

Minimización de costes (II)

Por lo tanto,

$$\boldsymbol{w}^T \sum_{n=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T = \sum_{n=1}^{N} t_n \boldsymbol{\phi}(\boldsymbol{x}_n)^T \implies \boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w} = \boldsymbol{\Phi}^T\boldsymbol{t} \quad (6)$$

$\boldsymbol{t} = (t_1, \ldots, t_N)^T$ y $\boldsymbol{\Phi}$ se llama *matriz de diseño* y viene dada por:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_0(\boldsymbol{x}_1) & \boldsymbol{\phi}_1(\boldsymbol{x}_1) & \ldots & \boldsymbol{\phi}_{M-1}(\boldsymbol{x}_1) \\ \boldsymbol{\phi}_0(\boldsymbol{x}_2) & \boldsymbol{\phi}_1(\boldsymbol{x}_2) & \ldots & \boldsymbol{\phi}_{M-1}(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\phi}_0(\boldsymbol{x}_N) & \boldsymbol{\phi}_1(\boldsymbol{x}_N) & \ldots & \boldsymbol{\phi}_{M-1}(\boldsymbol{x}_N) \end{bmatrix} \quad (7)$$

Ecuaciones normales

Y los pesos óptimos vienen dados por

$$\boldsymbol{w}^* = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{t} \tag{8}$$

Éstas se conocen como *ecuaciones normales* para el problema de mínimos cuadrados y proporcionan una expresión cerrada para determinar $\boldsymbol{w}^*$.

Finalmente, las predicciones se calculan como:

$$y(\boldsymbol{x}, \boldsymbol{w}^*) = (\boldsymbol{w}^*)^T\boldsymbol{\phi}(\boldsymbol{x}) \tag{9}$$

Métodos iterativos

El uso de las ecuaciones normales puede ser poco práctico en algunos casos:

- La inversión de la matriz $\mathbf{\Phi}^T\mathbf{\Phi}$ tiene complejidad $O(M^{2,8})$ (ver Straseen complejidad). Por lo tanto, puede ser costoso para valores de M grandes.
- Los datos de entrenamiento pueden no estar disponibles en un solo lote, sino secuencialmente y, por lo tanto, las ecuaciones normales no podrían usarse si se requieren predicciones *online*.

Por estas razones, es interesante disponer de algoritmos iterativos.

Optimización de descenso de gradiente

Los métodos iterativos se basan en el *descenso de gradiente*:

1. Establecer pesos iniciales para $\boldsymbol{w}$ (por ejemplo, al azar)
2. Repetir:

   $$\boldsymbol{w}^{(\text{new})} = \boldsymbol{w}^{(\text{old})} - \eta \nabla J(\boldsymbol{w}^{(\text{old})}) = \boldsymbol{w}^{(\text{old})} - \eta[\boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{w}^{(\text{old})} - \boldsymbol{\Phi}^T \boldsymbol{t}]$$

   hasta la convergencia

El parámetro η se llama *learning rate* y controla las propiedades de convergencia del algoritmo. Nótese que $\nabla J(\boldsymbol{w})$ ya se ha obtenido en la ec. (5) y aquí simplmente se ha reescrito en forma de matriz.

Overfitting

En la diapositiva 12 el polinomio de grado $M = 3$ parece el mejor, pero el coste $J(\boldsymbol{w}^*)$ es mínimo para $M = 9$ (¿por qué?). Esta situación se llama *overfitting*, y ocurre cuando la hipótesis no generaliza bien a nuevos puntos, incluso si encaja muy bien bien en los puntos del conjunto de entrenamiento. Una forma de corregir este problema es usar un conjunto de datos más extenso. Pero, ¿qué pasa si no hay más datos disponibles?

Se proporciona información adicional en la tabla a la derecha que muestra los pesos de los polinomios óptimos. Cuanto mayor sea el grado, mayores serán las fluctuaciones y, por lo tanto, los pesos.

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Regularization

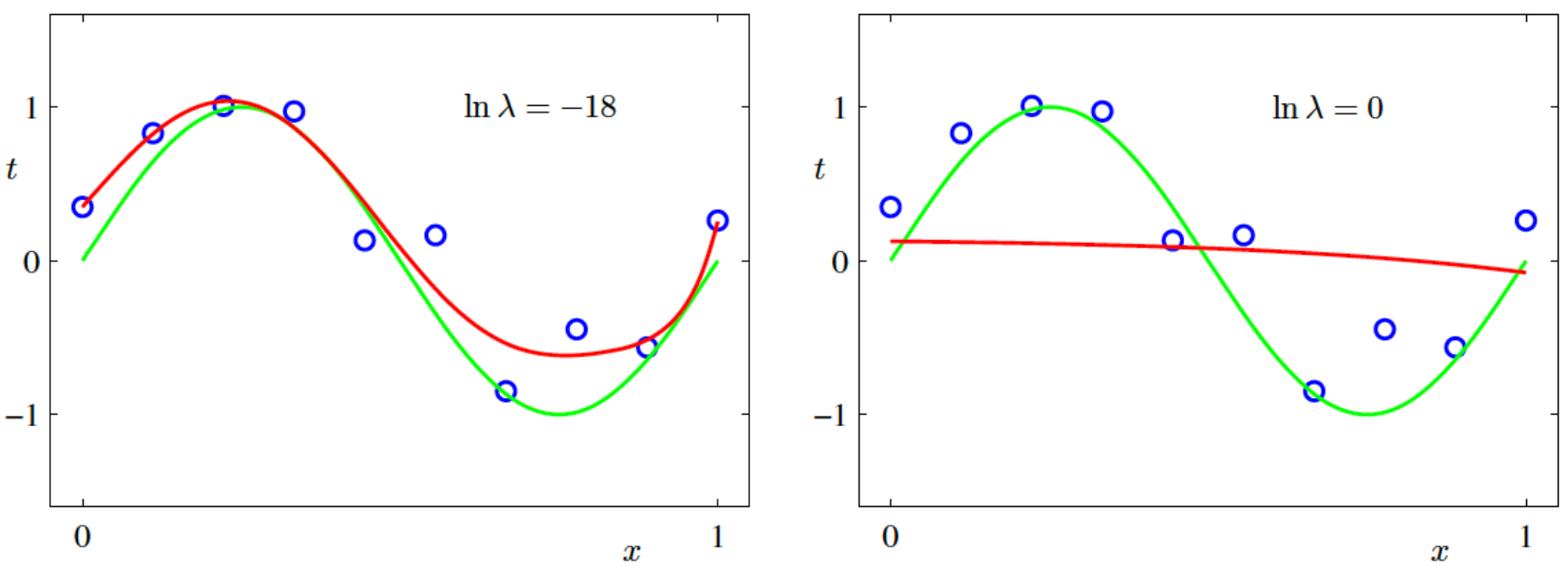
La última observación sugiere penalizar las soluciones con pesos grandes, lo cual se puede hacer agregando un *término de regularización* en la función de coste dada en la ecuación. (3), que conduce a:

$$\widetilde{J}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}[\sum_{j=0}^{M-1} w_j\phi_j(\boldsymbol{x}_n) - t_n]^2 + \frac{\lambda}{2}\sum_{j=0}^{M-1}|w_j|^q \qquad (10)$$

El caso de $q = 2$ se llama regresión *ridge*, y el segundo término en el lado derecho de la ecuación anterior se puede escribir como $\frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$. Tiene la ventaja de tener una solución cerrada:

$$\boldsymbol{w}^* = (\lambda I + \boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{t} \qquad (11)$$

Regularización (II)



Regresión ridge para dos parámetros de regularización λ aplicado a polinomios de $M = 9$ grados.

El caso de $q = 1$ se llama regresión *lasso*. Tiene la ventaja que si λ es suficientemente grande, algunos de los coeficientes w son forzados a cero, lo que conduce a un modelo disperso (*sparse model*) que selecciona qué funciones de la base no juegan ningún papel.

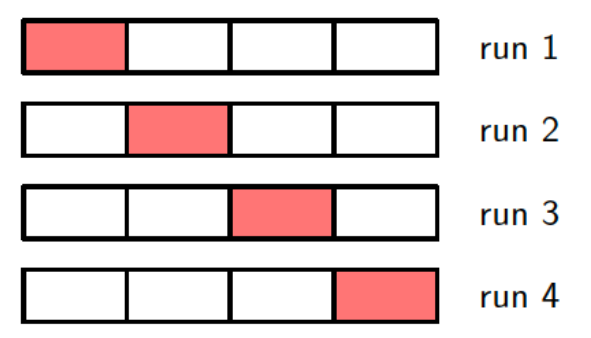
Validación cruzada

En una aplicación práctica, es necesario determinar el parámetro de regularización λ, así como otros parámetros (por ejemplo, los parámetros internos de la base de funciones). Estas variables, que condicionan la complejidad del modelo se denominan *hiper-parámetros*.

Una forma adecuada de determinarlos es usar *cross-validation*. La idea es separar del conjunto de entrenamiento una parte seleccionada al azar (conjunto de validación), y probar en él modelos entrenados con los datos de entrenamiento restantes. La S-fold cross-validation promedia el rendimiento al hacer S particiones independientes del conjunto de entrenamiento y probar los modelos entrenados contra el correspondiente conjuntos de validación.

Validación cruzada (II)

Ejemplo de S-fold cross-validation que divide el conjunto de datos en S partes de igual tamaño. El conjunto de validación para cada ejecución está en rojo.



Si el conjunto de datos es pequeño puede haber overfitting a los datos de validación. Habitualmente se separa un tercer conjunto (*test set*) en el que se evalúa el modelo seleccionado. Un inconveniente de la validación cruzada es que reduce los datos de entrenamiento disponibles.

Enfoque probabilístico

Enfoque probabilístico

En nuestro tratamiento anterior, hemos ignorado la naturaleza aleatoria del conjunto de entrenamiento donde el target tiene un componente ruidoso como se muestra en la curva de la diapositiva 12.

De hecho, se puede suponer que el objetivo es

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon$$

donde ϵ es una variable aleatoria Gaussiana de media cero con varianza β^{-1} (β se llama la *precisión*). Por lo tanto,

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}) = \mathcal{N}(t|\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1})$$

Enfoque probabilístico

Por lo tanto, dado $\boldsymbol{X} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_N}\}$ con N instancias obtenidas independientemente, su verosimilitud es:

$$p(t|\boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1})$$

Por lo tanto,

$$\ln p(t|\boldsymbol{X}, \boldsymbol{w}, \beta) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta\frac{1}{2}\sum_{n=1}^{N}[t_n - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x})]^2$$

El último término es $\beta J(\boldsymbol{w})$, y dado que $\boldsymbol{w}$ no aparece en los otros términos, su estimador de máxima verosimilitud (ML) bajo ruido gaussiano coincide con los pesos óptimos obtenidos a través del enfoque de ajuste de curva, es decir, $\widehat{\boldsymbol{w}}_{\text{ML}} = \boldsymbol{w}^*$.

Enfoque bayesiano

Enfoque bayesiano

La visión bayesiana considera la probabilidad como una medida de incertidumbre, a diferencia del paradigma clásico que es frecuentista. Es decir, los pesos del modelo se consideran variables desconocidas a las que se asigna una medida de probabilidad. Dado un conjunto de datos, la incertidumbre sobre los pesos se puede reducir utilizando la regla de Bayes:

Dadas las variables aleatorias a y b, y $p(a|b)$, entonces

$$p(b|a) = \frac{p(a,b)}{p(a)} = \frac{p(a|b)p(b)}{p(a)} \tag{12}$$

donde se obtiene $p(a)$ al marginalizar $p(a,b)$ sobre la distribución de b. $p(b)$ se llama distribución a priori de b y $p(b|a)$ distribución a posteriori de b dada la evidencia a.

Enfoque bayesiano (II)

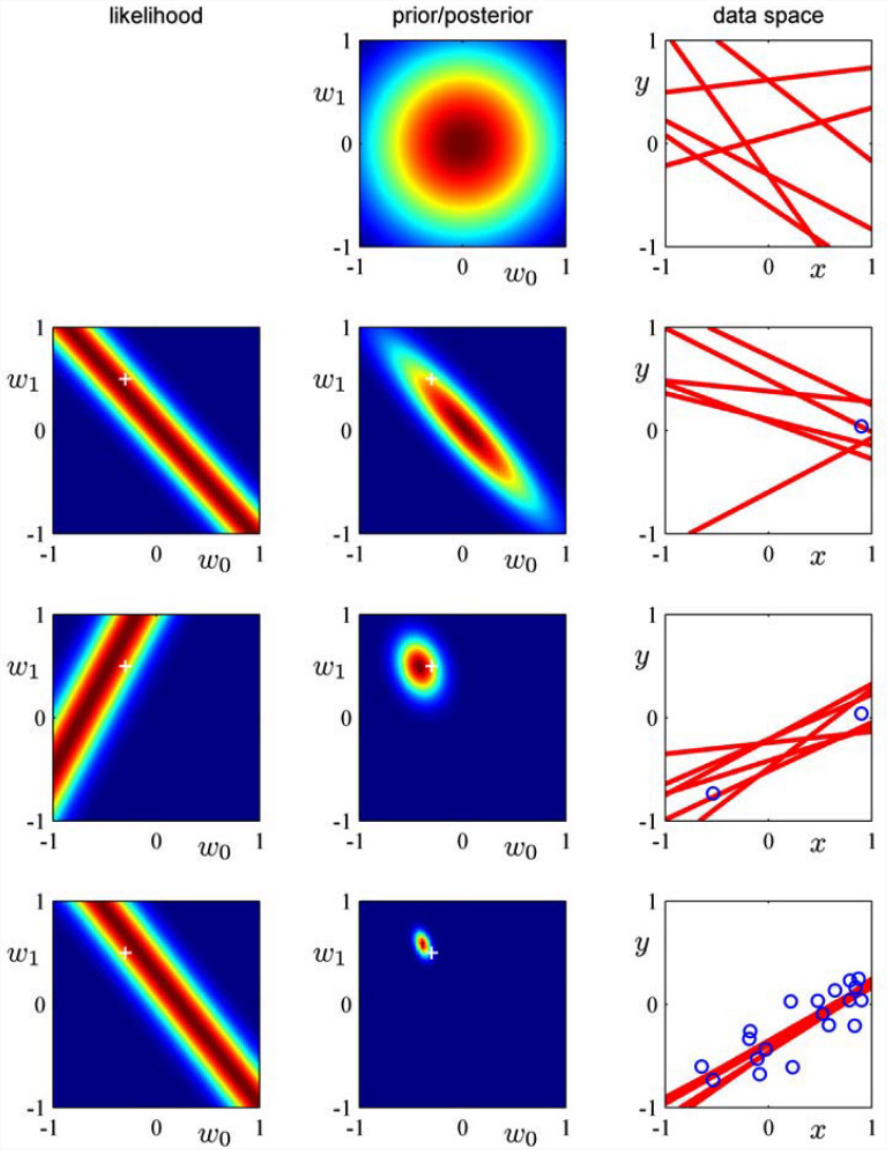
Por ejemplo, en nuestro problema de regresión lineal, estableciendo una distrución a priori $p(\boldsymbol{w}) = \mathcal{N}(0, \alpha^{-1} I)$, y dada la evidencia $\boldsymbol{X}, \boldsymbol{t}$, la distribución a posteriori de $\boldsymbol{w}$ es:

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{t}) = \mathcal{N}(\boldsymbol{w}|\beta \boldsymbol{S}_N \boldsymbol{\Phi}^T \boldsymbol{t}, \boldsymbol{S}_N)$$

donde $\boldsymbol{S}_N = (\alpha I + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1}$.

los estimadores máximo-posteriori (MAP) de $\boldsymbol{w}$ está dado por la maximización de la anterior expresión. Esta maximización coincide con la minimización del coste regularizado $\widetilde{J}(\boldsymbol{w})$ con $\lambda = \frac{\alpha}{\beta}$.

Enfoque bayesiano (III)



Ejemplo de aprendizaje bayesiano secuencial para un modelo lineal $y(x, \boldsymbol{w}) = w_0 + w_1 x$ con distribución del target $p(t|x, \boldsymbol{w}) = \mathcal{N}(w_0 + w_1 x, \beta^{-1} I)$.
La distribución a priori de $\boldsymbol{w}$ es gaussiana y cada nuevo punto (x, t) (círculo azul) tiene una probabilidad $p(t|x, \boldsymbol{w})$. Así, $p(\boldsymbol{w}|t) \propto p(t|x, \boldsymbol{w})p(\boldsymbol{w})$, cuya normalización permite obtener la distribución a posteriori. Esta distribución se convierte en la distribución a priori para la próxima iteración, y así sucesivamente. La columna de la derecha muestra funciones de hipótesis muestreadas de $\boldsymbol{w}$. A medida que hay nuevas pruebas (puntos) disponibles, la incertidumbre (en la distribución de $\boldsymbol{w}$) se reduce.

Enfoque bayesiano (IV)

En realidad, en lugar de en $\boldsymbol{w}$ estamos interesados en la predicción de la distribución $p(t|\boldsymbol{t})$ para nuevos valores de $\boldsymbol{x}$, que corresponde a la marginación de $p(t|\boldsymbol{t}, \boldsymbol{w})$ sobre el distribución de $\boldsymbol{w}$:

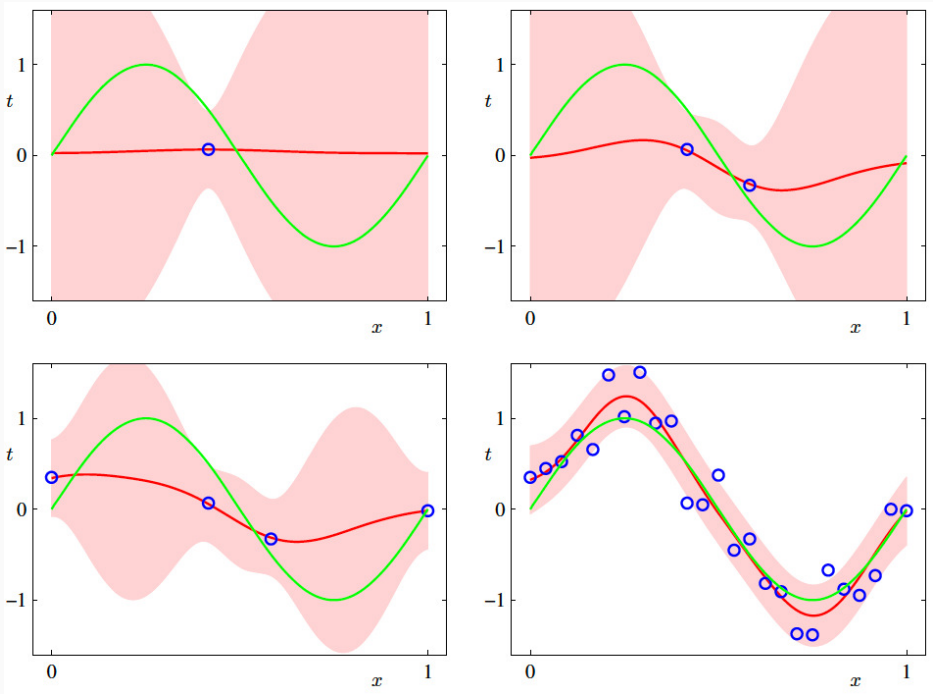
$$p(t|\boldsymbol{t}, \boldsymbol{w}) = \int p(t|\boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{t}) d\boldsymbol{w}$$

Al resolver la integral anterior (ver Bishop 3.3.2) se obtiene:

$$p(t|\boldsymbol{t}, \boldsymbol{w}) = \mathcal{N}(t|(\lambda I + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{t} \boldsymbol{\phi}(\boldsymbol{x}), \beta^{-1} + \boldsymbol{\phi}^T(\boldsymbol{x}) \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}))$$

El estimador MAP de t es gaussiano y su esperanza, como se ha indicado, coincide con el de la regresión regularizada con parámetro $\lambda = \frac{\alpha}{\beta}$

Enfoque bayesiano (V)



La distribución predictiva es importante porque permite construir intervalos de confianza para la predicción. En la figura se muestran las regiones de confianza que abarcan una desviación estándar a cada lado de la media. La base está compuesta por $M - 1 = 9$ funciones gaussianas.

Enfoque bayesiano (VI)

En el tratamiento por ajuste de curvas la idoneidad de una predicción debe ser evaluada viendo el rendimiento sobre un conjunto de validación. Con la distribución predictiva se pueden construir directamente regiones de confianza.

También es posible aplicar un tratamiento bayesiano que considere una mezcla de L modelos $\{\mathcal{M}_l\}$, cada uno definiendo su familia de parámetros correspondiente, e incluso considerando distribuciones a priori sobre α y β. Estudiantes interesados puede consultar Bishop 3.4 y 3.5.

Balance sesgo/varianza

La compensación de sesgo-varianza

Se puede obtener más información sobre la naturaleza de los modelos analizando la estructura del error en la función de coste. Nótese que el conjunto de datos en realidad puede considerarse aleatorio, y, por lo tanto, la pérdida asociada debe entenderse también como una variable aleatoria.

Por lo tanto, es posible calcular la esperanza de esta pérdida. (con respecto a la distribución aleatoria del conjunto de datos), resultando:

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

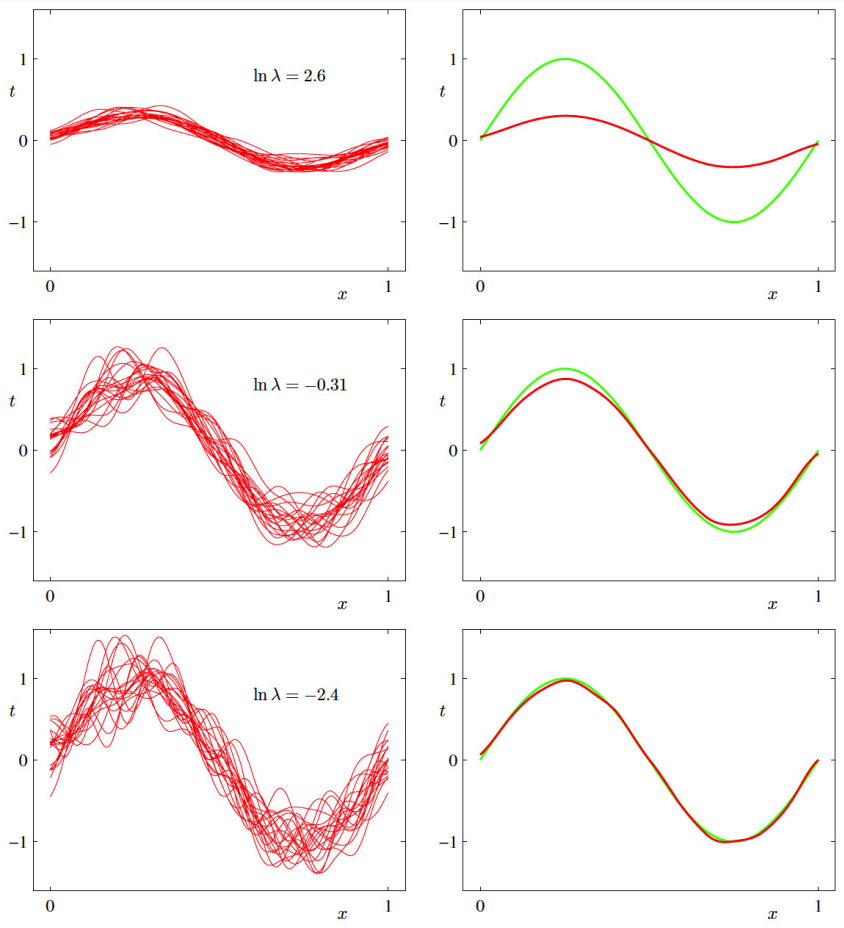
Tel intercambio de sesgo-varianza (II)

Estos términos están relacionados con diferentes causas:

- El sesgo se debe a la falta de flexibilidad del modelo, e.g. una base de funciones pequeña. Produce underfitting.
- Por otro lado, la varianza se debe al sobreajuste provocado por usar un modelo demasiado flexible.
- El noise es el error de la fuente de datos, y como tal, irreducible.

La complejidad (flexibilidad) del modelo está controlada por el parámetro de regularización λ. La solución óptima requiere un balance entre el sesgo y la varianza y corresponde con valores intermedios λ.

La compensación de sesgo-varianza (III)



Ejemplo de balance sesgo/varianza basado en 100 conjuntos de datos, cada uno con $N = 25$ puntos de datos. El modelo tiene $M = 25$ (24 funciones de base gaussianas y el parámetro de bias - no confundir con el sesgo estadístico). La columna izquierda muestra el resultado de ajustar el modelo a los conjuntos de datos para varios valores de $\ln \lambda$. La columna derecha muestra el promedio de los 100 ajustes (rojo) junto con La función sinusoidal a partir de la cual se generaron los conjuntos de datos (verde). La mejor opción es el valor intermedio de λ. Una λ alta reduce la complejidad del modelo, por lo que hará predicciones con poca varianza (pero su media está lejos de la curva real). Por otro lado, una λ pequeña permite modelos con alta complejidad, con varianza grande (aunque, la media de las predicciones casi coinciden con la curva original).