Lezione 5

Nicholas Attardo

April 2025

1 Regressione e valutazione del modello

1.1 Fasi del modello

1.1.1 Training

L'obiettivo primario del training è ottenere i parametri del modello θ. Durante questa fase, il modello apprende dalle relazioni presenti nel **set di training**.
L'algoritmo di training, come la discesa del gradiente **aggiorna iterativamente i parametri del modello** per minimizzare la **funzione di loss** $J(\theta)$.
La funzione di loss quantifica l'**errore tra le previsione del modello e i valori reali** nel training set.

Modelli e Funzione di Loss nel Training

Nella regressione lineare, la funzione del modello è il prodotto scalare tra il vettore dei parametri e il vettore di input.
$f_\theta(x) = v^t x = v_0 + v_1 x_1 + ... + v_d x_d$ dove xeR^d con $d >= 1$.
Nei modelli polinomiali, una funzione ϕ **trasforma i dati di input in uno spazio di dimensioni più elevate** prima di applicare un modello lineare.

La funzione di loss utilizzata, ad esempio, è l'errore quadratico medio (MSE) .
L'obiettivo del training è trovare i parametri θ che minimizzano questa funzione di loss . L'aggiornamento dei parametri avviene simultaneamente per tutte le componenti del vettore dei parametri, tenendo conto dei valori precedenti e di un tasso di apprendimento γ applicato al gradiente della funzione di loss rispetto ai parametri.

Regolarizzazione durante il Training

La regolarizzazione è un modo per prevenire l'**overfitting** durante il training. Le regolarizzazione consiste nell'**aggiungere un termine di penalizzazione alla funzione di loss**. Questo termine penalizza i valori elevati dei parametri, incoraggiando il modello a essere più semplice e a generalizzare meglio.

1.1.2 Valutazione

Questa fase segue il **training** del modello, durante il quale i parametri θ vengono appresi dai dati. L'obiettivo principale della valutazione è **misurare le prestazioni del modello addestrato** e la sua capacità di **generalizzare** a dati nuovi, non visti durante il training.

Quando avviene la Valutazione:
La valutazione avviene **dopo la fase di training**, una volta che i parametri del modello sono stati determinati. Per una valutazione affidabile, è cruciale utilizzare **dati distinti da quelli utilizzati per l'addestramento**.
Metriche di Valutazione:

- **Errore Quadratico Medio (MSE)**: Indicato come $J\theta$.
  Formula:

$$\frac{1}{m}\sum_{i}^{m}\left(f_{\hat{\theta}(x^{(i)})-g^{(i)}}\right)^2 \approx J(v)$$

  Questa metrica calcola la media dei quadrati degli errori tra le previsioni del modello e i valori reali.

- **Errore Assoluto Medio (MAE):** Formula:

$$\frac{1}{m}\sum_{i=1}^{m}\left|f_\theta(x^{(i)}) - y^{(i)}\right|$$

  Questa metrica fornisce la media del valore assoluto degli errori.

- **Root Mean Squared Error (RMSE):** Definito come la radice quadrata dell'MSE e rappresenta la deviazione standard degli errori. Fornisce informazioni sulla dispersione degli errori attorno alla media.

La scelta delle metrica dipende dal **task specifico** e dalla **semantica** che si vuole misurare. Ad esempio, l'MAE può essere più interpretabile per problemi come la stima del numero delle macchine. L'obiettivo generale è che queste misure di errore siano **il più piccole possibile** per indicare un buon modello.

1.2 REC Curve

La **REC Curve** è un metodo grafico per valutare i modelli di regressione nel contesto più ampio della valutazione del modello.

Costruzione e Interpretazione della REC Curve
La costruzione di una REC Curve segue un processo sistematico:

1. **Calcolo degli errori:** Per ogni campione nel test set, si calcola l'errore tra il valore predetto e il valore effettivo.
2. **Ordinamento degli errori:** Gli errori calcolati vengono ordinati in modo crescente
3. **Costruzione del grafico:**
   Asse x: Rappresenta la tollerenza all'errore ϵ'_i.
   La tolleranza di errore è un valore che fissiamo arbitrariamente per definire quanto siamo disposti a "tollerare" come errore massimo.
   Esempio: Abbiamo come valori di errore assoluto ϵ' :
   $10.000, 5.000, 20.000, 10.000, 10.000$. Possiamo prendere come valori da plottare nell'asse $x : 5.000, 10.000, 20.000$
   Asse y: Mosta la percentuale cumulativa di punti predetti entro la tolleranza ϵ, calcolata come $P(\epsilon) = \frac{k}{m}$ dove k è il numero di errori minori o uguali a ϵ
   Esempio:
   Entro la tolleranza di 5.000 abbiamo un solo valore, quindi avremo come valore di $P(\epsilon) = \frac{1}{5}$
4. **Tracciamento della curva:** La curva viene costruita interpolando i punti $(\epsilon_i, P(\epsilon_i))$, risultando in una funzione monotona crescente che parte da $(0, 0)$ e raggiunge $(\epsilon_m, 1)$, dove ϵ_m è l'errore massimo osservato.

Per costruire una REC Curve, si prendono i **valori di errore** ottenuti dal modello su un set di test. Questi errori, calcolati per ogni campione di test (X test), possono essere, ad esempio, il MSE.
Questi errori vengono poi **ordinati in maniera crescente**
Sull'asse orizzontale del grafico si rappresenta l'errore ordinato $(\epsilon'_0, ..., \epsilon'_m)$
Sull'asse verticale si traccia la **percentuale cumulativa di errori minori o uguali** al valore corrispondente sull'asse orizzontale.
Formalmente il valore P corrispondente sull'asse verticale è dato da :
$P = \frac{K}{m}$
dove K non è altro che il numero normalizzato di elementi x^i
Questo valore di P varia da 0 a 1 man mano che ci si sposta lungo l'asse orizzontale degli errori. Una REC Curve ideale si alza rapidamente verso 1, indicando che la maggior parte degli errori commessi dal modello sono piccoli Questo si calcola come il numero di errori minori o uguali a una certa soglia (ϵ'_g) diviso per il numero totale di campioni di test (n o m)
La curva risultante parte da zero e si alza progressivamente fino a raggiungere il valore massimo di 1.

Valutazione del Modello tramite REC Curve

Un modello migliore avrà una REC Curve che si alza velocemente verso 1.Questo indica che la maggior parte degli errori commessi dal modello sono piccoli.
L'**area sotto la curva** fornisce una misura complessiva della bontà dell'algoritmo. Un'area maggiore indica un algoritmo migliore.
Abbiamo una lista ordinata dal valore di errore più piccolo al valore di errore più grande. Ciò implica che se il primo valore è posto nel plot in una posizione con ordinata prossima a 1, significa che il nostro modello si comporta bene.

Confronto tra Modelli:

La REC Curve è utile per **confrontare diversi modelli di regressione** sullo stesso set di test.
Permette di andare oltre la semplice comparazione di metriche di errore medio (come MSE o MAE). Due modelli potrebbero avere errori medi simili, ma la REC Curve può rivelare differenze nella distribuizione degli errori, mostrando quale modello commette più errori piccoli e meno errori grandi.

Contesto della Valutazione del Modello

La REC Curve viene applicata nella fase di **valutazione del modello**, dopo che il modello è stato addestrato (sul training set) e potenzilalmente ottimizzato (utilizzando un validation set).
Si utilizza il **set di test** per ottenere una stima imparizale delle prestazioni del modello su dati mai visti.
La REC Curve. insieme ad altre metriche di errore (MSE, MAE, RMSE), fornisce una visione più completa delle capacità predittive del modello di regressione.

Differenza della ROC Curve

Esiste una curva simile chiamata **ROC Curve** che viene utilizzara per la **valutazione di modelli di classificazione.** mentre la REC Curve è specifica per la regressione.

1.3 Equazione Normale

L'algoritmo della discesa del gradiente trova i parametri che minimizzano la funzione costo $J(\theta)$ in maniera iterativa.
Il metodo dell'equazione normale trova i parametri ottimali θ in un singolo step risolvendo analiticamente il problema:

$$T = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & x_2^{(1)} & ... & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & ... & x_n^{(2)} \\ x_0^{(3)} & x_1^{(3)} & x_2^{(3)} & ... & x_n^{(3)} \\ ... & ... & ... & ... & ... \\ x_0^{(m)} & x_1^{(m)} & x_2^{(m)} & ... & x_n^{(m)} \end{bmatrix} \; y = \begin{bmatrix} y^{(1)} \\ ... \\ y^{(m)} \end{bmatrix}$$

Dato un set di training T, una matrice dei dati X (con m set di dati e n+1 colonne, inclusa una colonna per l'intercetta) e un vettore di output Y, l'**Equazione Normale** fornisce una formula diretta per calcolare il vettore dei parametri θ che minimizza la funzione di costo.

$$\hat{\theta} = min_\theta J(\theta) = (T^T T)^{-1} T^T y$$

Dove X^T è la trasposta della matrice X, $(T^T T)^{-1}$ è l'inversa della matrice $T^T T$, e y è il vettore dei valori target.
Il vettore θ risultante avrà una dimensione di :

$$T^T T \to [(n+1) * m] * [m * (n+1)] \to (n+1) * (n+1)$$
$$(T^T T)^{-1} T^T \to (n+1) * (n+1) * (n+1) * m \to (n+1) * m$$
$$(T^T T)^{-1} T^T * y \to [(n+1) * m] * [m * 1] \to (n+1) * 1 \to \begin{bmatrix} \theta_0 \\ ... \\ \theta_n \end{bmatrix}$$

$n+1$, corrispondente ai n parametri più il termine di intercetta θ_0.
$T^T T$ deve essere invertibile (ovvero deve esistere una matrice che moltiplicata per questa produce una matrice identità). Se il numero di feature n è molto maggiore del numero di campioni di training m questa matrice potrebbe non essere invertibile.

GRADIENT DESCENT VS EQUAZIONE NORMALE

- **Discesa del Gradiente**: Si deve scegliere γ, necessita di tante iterazioni, funziona anche quando n è grande.
- **Equazione normale**: Nessun γ da scegliere, non è un metodo iterativo, bisogna computare $O(n^3)$, molto lento se n è molto grande.

Preferiamo l'equazione normale per valori di n piccoli come fino a 1000 unità, dopo preferiamo la discesa del gradiente.

1.4 Capacità del modello

La capacità del nostro modello deve essere in relazione alla sua **complessità**, spesso illustrato attraverso l'uso di **regressione polinomiale**. Un modello con maggiore capacità è in grado di rappresentare relazioni più complesse nei dati. Ad esempio, incrementando il grado di un polinomio in un modello di regressione, si aumenta la sua capacità di adattarsi ai dati di training.
Se il modello che utilizziamo come ipotesi è molto complesso (es. polinomiale di grado alto) rispetto ai dati da fittare allora si può verificare:

- Basso errore sul training set
- Alto errore sul validation/test, il modello non generalizza su nuovi esempi mai visti prima, saremo in presenza di **overfitting**!

Underfitting
Modello **non è in grado di generalizzare sui campioni**, manifestandosi con un **errore alto sia sul training set che sul test set (o validation set)**. Questa condizione è associata a un **modello troppo semplice** rispetto alla complessità dei dati.
Nello specifico, è presentato uno scenario con tre modelli di complessità crescente fittati agli stessi dati di training. Il modello più semplice mostra un **errore alto sul training set** e, pur approssimando meglio un nuovo punto ntel test rispetto al modello più complesso, comunque presenta un errore nel test. Questo illustra come un modello con **capacità insufficiente** (cioè un modello troppo semplice) non riesca a catturare la vera relazione sottostante nei dati, portando a un **bias alto**.
L'**underfitting** è la situazione in cui il modello **non è capace di generalizzare sui campioni** e si verificano entrambe le condizioni di errore alto sul training e sul test.
In caso di unferfitting, quando si utilizza un modello troppo semplice rispetto ai dati, si osserva un **errore alto sia sul training (J_{train}) che sul validation set J_{val}.** Le curve ipotetiche degli errori in funzione della complessità del modello (grado del polinomio) mostrano come, per modelli troppo semplici (basso grado del polinomio), sia l'errore sul training che quello sul validation rimangono elevati.
In presenza di **bias (underfitting)**, aumentare il numero di campioni di training generalmente non aiuta a migliorare le prestazioni. Questo perchè il modello è intrinsecamente limitato nella sua capacità di apprendere la relazione sottostante a causa della sua semplicità.
Per affrontare l'**underfitting** aumentiamo la **complessità** del modello. Ad esempio, se un modello lineare è in underfitting per dati non lineari, si potrebbe considerare l'utilizzo di un modello polinomiale di grado superiore per incrementare la capacità del modello di fittare relazioni più complesse. Per far scendere l'errore in caso di underfitting, è necessario aumentare la complessità del modello.

Overfitting

L'overfitting è associato a una **varianza alta**. Modelli in overfitting sono molto sensibili ai piccoli cambiamenti nei dati di training, portando a modelli molto diversi anche con variazioni minime nei dati.
L'overfitting si manifesta quando un modello fa benissimo nel training ma fa malissimo nel test. Un modello complesso può fittare i dati di training in modo quasi perfetto, ma questa aderenza ai dati di training include anche il rumore presente in essi. Di conseguenza, il modello non riesce a generalizzare bene a dati mai visti. Più il modello è complesso, più la forma del polinomio può risultare irregolare. Inoltre, il modello è fortemente dipendente dai dati: rimuovendo anche un solo elemento e rieseguendo il training, si otterrebbe un risultato completamente diverso.
Per affrontare il problema dell'overfitting, si può **diminuire la complessità del modello** o utilizzare tecniche di **regolarizzazione**. Inoltre, l'**aumento del numero di campioni di training può aiutare** a migliorare le prestazioni in caso di overfitting, fornendo al modello più dati da cui generalizzare e riducendo l'impatto del rumore specifico del training set.

1.5 Regolarizzazione

La **regolarizzazione** è una tecnica fondamentale utilizzata per prevenire l'**overfitting** e semplificare il modello durante la fase di training. L'obiettivo principale della regolarizzazione è migliorare la capacità del modello di generalizzare a dati non visti, evitando che si adatti troppo strettamente ai dati di training, inclusi rumore e variazioni casuali.

1.5.1 Il Ruolo della Funzione di Loss Regolarizzata

Nel contesto della regolarizzazione, la funzione di loss originale ($J(\theta)$), che misura l'errore sui dati di training, viene modificata aggiungendo un **termine di penalizzazione**. Questo termine è progettato per penalizzare i valori elevati dei parametri del modello (θ_j), incentivando l'algoritmo di ottimizzazione a mantenere i pesi più piccoli possibile. La nuova funzione di loss diventa:

$$J_{\text{reg}}(\theta) = J(\theta) + \lambda \sum_{j=1}^{d} \theta_j^2$$

dove:
- $J(\theta)$ è la **data loss**, che quantifica l'errore del modello sui dati di training.
- λ è un **iperparametro** che controlla l'intensità della regolarizzazione.
- $\sum_{j=1}^{d} \theta_j^2$ è il termine di regolarizzazione, che penalizza i quadrati dei parametri associati alle feature $(\theta_1, \theta_2, ..., \theta_d)$.

È importante notare che il parametro θ_0 (il bias o intercetta) **non è incluso** nella sommatoria. Questa scelta è motivata dal fatto che θ_0 rappresenta il termine che permette di spostare liberamente l'intercetta del modello, garantendo un migliore adattamento ai dati senza forzare l'intercetta verso zero.

1.5.2 Regolarizzazione L1 e L2

Esistono due tipi comuni di regolarizzazione:

- **Regolarizzazione L2 (Ridge)**: Utilizza il quadrato dei parametri (θ_j^2). Penalizza maggiormente i valori dei parametri più alti, distribuendo uniformemente i pesi e riducendoli gradualmente.

- **Regolarizzazione L1 (Lasso)**: Utilizza il valore assoluto dei parametri ($|\theta_j|$). Favorisce modelli "sparsi", ovvero modelli in cui alcuni pesi vengono ridotti esattamente a zero, eliminando feature irrilevanti.

Entrambe le norme sono efficaci per prevenire l'overfitting, ma la scelta dipende dal contesto e dagli obiettivi del modello.

1.5.3 Il Ruolo dell'Iperparametro λ

L'iperparametro λ gioca un ruolo cruciale nel bilanciare due obiettivi contrastanti:

1. Minimizzare l'errore sui dati di training (**data loss**).

2. Mantenere i pesi del modello di piccola entità (**penalizzazione**).

Un valore **elevato di** λ impone una penalizzazione maggiore sui pesi elevati, portando a un modello più semplice. Tuttavia, un valore troppo alto può causare **underfitting**, poiché il modello diventa incapace di catturare le relazioni sottostanti nei dati. Al contrario, un valore **basso di** λ riduce l'impatto del termine di regolarizzazione, lasciando il modello più flessibile. Tuttavia, un valore troppo basso può portare all'**overfitting**, poiché il modello si adatta troppo strettamente ai dati di training, incluso il rumore.

La scelta del valore ottimale di λ è fondamentale ed è solitamente effettuata tramite validazione incrociata. Si addestrano diversi modelli con valori diversi di λ, mantenendo fissi gli altri iperparametri, e si seleziona il valore che produce le migliori prestazioni su un **validation set**.

1.5.4 Effetti della Regolarizzazione

La regolarizzazione ha due effetti principali:

1. **Prevenzione dell'Overfitting**: Limitando la grandezza dei parametri, il modello diventa meno flessibile e meno incline ad adattarsi al rumore specifico dei dati di training. Questo migliora la sua capacità di generalizzare su dati nuovi.

2. **Semplificazione del Modello**: Riducendo i pesi delle feature irrilevanti, il modello diventa più semplice e interpretabile. Nel caso della regolarizzazione L1, alcune feature possono essere completamente eliminate, portando a modelli più sparsi.

1.5.5 Sintesi

In sintesi, la regolarizzazione è uno strumento essenziale per ottenere modelli robusti e generalizzabili. Aggiungendo un termine di penalizzazione alla funzione di loss, la regolarizzazione incentiva l'algoritmo di ottimizzazione a mantenere i pesi di piccola entità, prevenendo l'overfitting e semplificando il modello. La forza della penalizzazione è controllata dall'iperparametro λ, che deve essere opportunamente sintonizzato tramite validazione. Scegliere tra regolarizzazione L1 e L2 dipende dagli obiettivi del modello, con la L1 che favorisce la selezione delle feature e la L2 che distribuisce uniformemente i pesi.

1.6 Selezione del Modello

La selezione del modello è un passo cruciale nel processo di machine learning, con l'obiettivo di individuare il modello che non solo si adatta bene ai dati di training, ma che è anche in grado di generalizzare efficacemente a dati non visti. Tra le tecniche utilizzate, l'**errore minimo sul validation set** rappresenta un criterio chiave per questa scelta. Attraverso un processo strutturato di addestramento, valutazione e confronto, è possibile identificare il modello che offre il miglior equilibrio tra complessità e capacità di generalizzazione.

Il processo di selezione del modello prevede diversi passaggi. Inizialmente, si considerano diversi modelli $(F_1, F_2, \ldots, F_k)$ che possono variare in termini di complessità. Ad esempio, si possono esplorare modelli polinomiali di grado crescente, da lineari (F_1) a più complessi (F_k). Per ciascun modello, si esegue una fase di training sui dati di training utilizzando una **funzione di loss regolarizzata** $(J_{\text{reg}}(\theta))$, che include un termine di regolarizzazione per mitigare il rischio di overfitting. Questo processo produce un set di parametri stimati $(\hat{\theta})$ per ogni modello.

Successivamente, i parametri stimati vengono utilizzati per fare previsioni sui dati del **validation set**, e si calcola l'errore $(J(\theta))$ per ciascun modello. È importante sottolineare che l'errore calcolato sul validation set **non include il termine di regolarizzazione** utilizzato durante il training. L'obiettivo di questa valutazione è misurare la **capacità di generalizzazione** di ciascun modello a dati nuovi, senza penalizzare ulteriormente la complessità. Gli errori ottenuti sul validation set per i diversi modelli vengono quindi confrontati, e il modello che produce l'errore più basso $(\arg\min J_{\text{val}})$ viene selezionato come il **modello migliore**. Questo modello è ritenuto quello con la maggiore probabilità di generalizzare bene a dati non visti.

Questo approccio è strettamente legato alla gestione dei problemi di **underfitting** e **overfitting**. Osservando l'errore sul training set (J_{train}) e sul validation set (J_{val}) per diversi modelli di complessità crescente, è possibile identificare situazioni critiche:

- **Underfitting**: Se un modello è troppo semplice, si osserverà un errore alto sia sul training set che sul validation set. Questo indica che il modello non riesce a catturare adeguatamente la struttura dei dati.

- **Overfitting**: Se un modello è troppo complesso, si osserverà un errore

basso sul training set ma un errore significativamente più alto sul validation set. Questo indica che il modello ha imparato troppo bene i dettagli e il rumore dei dati di training, perdendo la capacità di generalizzare.

L'obiettivo della selezione del modello è trovare un punto in cui l'errore sul validation set è minimo, indicando un buon equilibrio tra la capacità del modello di adattarsi ai dati di training e la sua capacità di generalizzare.

Lo stesso approccio di valutazione sul validation set può essere utilizzato anche per la selezione dell'iperparametro di regolarizzazione (λ). Si provano diversi valori di λ, si addestrano i modelli corrispondenti e si valuta l'errore sul validation set. Il valore di λ che minimizza l'errore sul validation set viene selezionato come il **valore ottimale**. Questo processo evidenzia come l'errore sul validation set sia uno strumento fondamentale per la selezione sia della **struttura del modello** che dei suoi **iperparametri**.

Infine, dopo aver selezionato il modello migliore basandosi sull'errore minimo sul validation set, è possibile valutare la performance finale del modello sul **test set**. Il test set fornisce una stima imparziale della capacità di generalizzazione del modello su dati completamente nuovi. È importante utilizzare il test set solo nella fase finale, per evitare che influenzi il processo di selezione del modello.

In sintesi, la selezione del modello tramite l'errore sul validation set è un processo robusto e versatile. Attraverso l'addestramento di diversi modelli, la valutazione delle loro prestazioni su un validation set e la scelta del modello con le migliori prestazioni di generalizzazione, è possibile identificare il modello che meglio bilancia complessità e capacità di generalizzare. Questo approccio, unito alla selezione degli iperparametri tramite lo stesso criterio, garantisce un processo di ottimizzazione completo ed efficace.

1.7 Curve di Errore vs Grado del Polinomio

Nel contesto della regressione polinomiale, l'analisi delle curve di errore rispetto al grado del polinomio è uno strumento fondamentale per la selezione del modello. Le curve di errore sul training set (J_{train}) e sul validation set (J_{val}) forniscono indicazioni cruciali sul comportamento del modello e permettono di diagnosticare fenomeni come **underfitting** e **overfitting**.

L'errore sul training set (J_{train}) tende a diminuire monotonicamente all'aumentare del grado del polinomio. Questo fenomeno è spiegabile con il fatto che modelli più complessi hanno una maggiore capacità di adattarsi ai dati di training. Tuttavia, questa riduzione continua dell'errore di training non garantisce necessariamente una buona generalizzazione a dati non visti. Al contrario, l'errore sul validation set (J_{val}) segue tipicamente una **curva a forma di U**, riflettendo un comportamento più complesso e informativo.

La curva di J_{val} può essere suddivisa in tre fasi principali:

1. **Underfitting (Gradi Bassi)**: Per gradi di polinomio bassi, sia J_{train} che J_{val} sono elevati. Questo indica una situazione di **underfitting**, in cui il modello è troppo semplice per catturare la struttura sottostante nei dati.

In questa fase, il modello non riesce a descrivere adeguatamente né i dati di training né quelli di validation.

2. **Equilibrio Ottimale (Gradi Intermedi)**: All'aumentare del grado del polinomio, J_{train} continua a diminuire, mentre J_{val} inizialmente diminuisce anch'esso. In questa fase, il modello diventa più capace di generalizzare, trovando un buon equilibrio tra adattamento ai dati di training e capacità di predire nuovi dati. Il punto minimo della curva di J_{val} rappresenta il **grado ottimale del polinomio**, corrispondente al modello con la migliore capacità di generalizzazione.

3. **Overfitting (Gradi Elevati)**: Se il grado del polinomio aumenta ulteriormente, J_{train} continuerà a diminuire fino a diventare molto basso, ma J_{val} inizierà ad aumentare significativamente. Questo indica una situazione di **overfitting**, in cui il modello si adatta troppo ai dettagli e al rumore specifico dei dati di training, perdendo la capacità di generalizzare bene a nuovi dati.

L'obiettivo nella selezione del modello è identificare il grado del polinomio che corrisponde al punto minimo della curva di J_{val}. Questo punto rappresenta il modello che si presume abbia la migliore capacità di generalizzazione a dati non visti. La differenza tra J_{train} e J_{val} fornisce ulteriori indizi sul comportamento del modello:

- Quando J_{val} è molto maggiore di J_{train}, è un chiaro segnale di overfitting, suggerendo che si dovrebbe considerare un modello meno complesso (grado del polinomio inferiore).

- Quando sia J_{val} che J_{train} sono alti, si è in una situazione di underfitting, e potrebbe essere utile considerare un modello più complesso (grado del polinomio superiore).

In sintesi, l'analisi delle curve di errore rispetto al grado del polinomio è cruciale per la selezione della giusta complessità del modello. Osservando come J_{train} e J_{val} si comportano al variare del grado del polinomio, è possibile diagnosticare problemi di underfitting e overfitting e scegliere il modello che minimizza l'errore di generalizzazione. Questo approccio garantisce un buon equilibrio tra la capacità del modello di adattarsi ai dati di training e la sua capacità di generalizzare a nuovi dati, garantendo prestazioni robuste e affidabili.

1.8 Curve di Errore vs Parametro di Regolarizzazione γ

Nell'ambito dell'ottimizzazione dei modelli di apprendimento supervisionato, il parametro di regolarizzazione γ gioca un ruolo cruciale nel bilanciare la complessità del modello e la sua capacità di generalizzazione. Le curve di errore, rappresentate dalle metriche J_{train} (errore sul training set) e J_{val} (errore sul validation set), forniscono un quadro completo dell'impatto di γ sulle prestazioni del modello.

L'errore di training J_{train}, calcolato senza includere la componente di regolarizzazione, tende a crescere all'aumentare di γ. Questo fenomeno è dovuto al

fatto che valori maggiori di γ introducono una penalizzazione più forte sui pesi dei parametri durante l'ottimizzazione, costringendo il modello a utilizzare pesi di valore più piccolo. Di conseguenza, il modello diventa più semplice e meno capace di adattarsi perfettamente ai dati di training, aumentando J_{train}. Al contrario, per valori di γ molto bassi, la penalizzazione sui pesi è trascurabile, permettendo al modello di diventare più complesso e di adattarsi eccessivamente ai dati di training. Tuttavia, questa maggiore complessità può portare a un aumento del rischio di overfitting, compromettendo la capacità del modello di generalizzare su nuovi dati.

Per valutare la capacità di generalizzazione, si ricorre all'errore sul validation set J_{val}, calcolato anch'esso senza il termine di regolarizzazione. La curva di J_{val} in funzione di γ assume tipicamente una forma a U. Per valori di γ molto piccoli, il modello tende all'overfitting, con J_{val} elevato a causa della scarsa generalizzazione. All'aumentare di γ, la regolarizzazione riduce l'overfitting, migliorando la capacità del modello di generalizzare e facendo diminuire J_{val}. Si raggiunge quindi un valore ottimale di γ che minimizza J_{val}, rappresentando il miglior compromesso tra bias e varianza. Tuttavia, se γ continua ad aumentare oltre questo punto, il modello diventa troppo semplice, portando a una situazione di underfitting, con conseguente aumento di J_{val} e J_{train}.

La selezione del valore ottimale di γ si basa sull'analisi delle curve di errore. In particolare, si esegue il training del modello per diversi valori di γ e si calcola J_{val} per ciascuno di essi. Il valore di γ che produce il J_{val} minimo viene scelto come il miglior parametro di regolarizzazione. Questo processo permette di diagnosticare eventuali problemi di overfitting (quando J_{val} è significativamente maggiore di J_{train} per γ piccolo) o underfitting (quando entrambi J_{val} e J_{train} sono elevati per γ grande). L'obiettivo finale è identificare la regione in cui J_{val} è minimo, garantendo una buona capacità predittiva del modello su nuovi dati.

In sintesi, l'analisi delle curve di errore in funzione di γ rivela il trade-off fondamentale tra overfitting e underfitting e guida la scelta di un modello che bilancia complessità e generalizzazione. Questo approccio costituisce un elemento chiave nella progettazione di modelli di machine learning robusti ed efficaci.

1.9 Curve di Errore vs Numero di Campioni (m)

L'analisi delle curve di errore in funzione del numero di campioni di training (m) rappresenta uno strumento fondamentale per comprendere il comportamento dei modelli di apprendimento automatico e diagnosticare eventuali problemi legati al *bias* (*underfitting*) e alla *varianza* (*overfitting*). Le curve di errore forniscono informazioni preziose su come l'errore di training (J_{train}) e l'errore di validazione (J_{val}) evolvono all'aumentare della quantità di dati disponibili, permettendo di valutare la capacità del modello di apprendere e generalizzare.

Comportamento Generale delle Curve di Errore

Per un modello fissato, si osserva che l'errore di training (J_{train}) tende inizialmente a crescere leggermente all'aumentare di m, poiché il modello fatica ad adattarsi perfettamente a un numero crescente di dati. Successivamente, J_{train} si stabilizza su un valore costante, riflettendo il limite intrinseco della capacità del modello di apprendimento. Al contrario, l'errore di validazione (J_{val}) diminuisce all'aumentare di m, poiché il modello impara meglio la distribuzione sottostante dei dati e migliora la sua capacità di generalizzazione. Questo andamento decrescente di J_{val} si stabilizza anch'esso dopo un certo numero di campioni, indicando che ulteriori dati hanno un impatto marginale sulle prestazioni.

Underfitting (Alto Bias)

L'*underfitting* si verifica quando un modello è troppo semplice per catturare la complessità della relazione sottostante nei dati. In questa situazione, sia J_{train} che J_{val} risultano elevati, evidenziando una scarsa capacità del modello di apprendere dai dati di training e di generalizzare a nuovi campioni. Un punto cruciale è che aumentare il numero di campioni di training (m) non migliora significativamente le prestazioni in caso di underfitting. L'errore elevato non dipende dalla quantità di dati, ma dalla limitata capacità del modello. La soluzione più efficace consiste nell'aumentare la complessità del modello, ad esempio utilizzando modelli più flessibili o aggiungendo feature rilevanti.

Overfitting (Alta Varianza)

L'*overfitting* si verifica quando un modello è troppo complesso e si adatta eccessivamente ai dati di training, memorizzando anche il rumore presente in essi. In questo caso, J_{train} è basso, mentre J_{val} risulta significativamente più alto, indicando una scarsa capacità di generalizzazione. Un modello con alta varianza è altamente sensibile ai dettagli specifici del set di training, e cambiamenti minimi nei dati possono portare a modelli completamente diversi. A differenza dell'underfitting, l'aggiunta di più campioni di training (m) può migliorare le prestazioni in caso di overfitting. L'aumento di m rende più difficile per il modello memorizzare il rumore, portando a una riduzione di J_{val} e a una migliore generalizzazione. Tuttavia, è importante notare che l'efficacia di questa strategia dipende dalla natura del problema e dalla complessità del modello. In alternativa, si possono applicare tecniche di regolarizzazione o ridurre la complessità del modello per mitigare l'overfitting.

Conclusione

In sintesi, l'analisi delle curve di errore rispetto al numero di campioni di training (m) fornisce una prospettiva chiara sul comportamento del modello e sui possibili problemi di bias e varianza. L'andamento di J_{train} e J_{val} deve essere considerato congiuntamente per diagnosticare correttamente se un modello soffre di underfitting o overfitting. Mentre l'aumento della quantità di dati è

una strategia efficace per ridurre la varianza in caso di overfitting, non risolve i problemi di bias associati all'underfitting. In quest'ultimo caso, è necessario intervenire sulla complessità del modello per migliorarne le prestazioni. Pertanto, l'interpretazione accurata delle curve di errore è essenziale per guidare le decisioni sull'ottimizzazione del modello e migliorare la sua capacità di generalizzazione.