**Riorganizzazione degli appunti 29/11**

**1. Cross Validation (CV)**

La **Cross Validation** è una tecnica per affrontare il problema di determinare la complessità di un modello o scegliere i suoi iperparametri prima di vedere i dati effettivi, aiutando a prevenire l'**overfitting**.

**Punti chiave:**

1. **Problema principale**:
   * Difficoltà nel scegliere la complessità ottimale di un modello.
   * La scelta dipende dal numero di parametri e iperparametri dell'algoritmo.
2. **Principio della CV**:
   * Riserva una parte dei dati di training come **validation set** per valutare le prestazioni predittive del modello.
   * Simula il comportamento del modello su dati non osservati.
3. **Divisione dei dati**:
   * **Training set**: addestrare il modello.
   * **Validation set**: testare le predizioni del modello.
4. **Iteratività**:
   * La CV viene ripetuta alternando il ruolo di ciascun esempio nei set di training e validation.
   * Il modello scelto ha le migliori prestazioni medie su tutte le iterazioni.

**Collegamento con altri temi:**

* **Overfitting**: adattamento eccessivo ai dati di training.
* **Bias-Variance Tradeoff**: compromesso tra modelli troppo semplici (bias alto) e troppo complessi (varianza alta).
* **Errori di predizione**: metriche come l'errore quadratico medio (MSE).
* La CV aiuta a scegliere un modello bilanciato tra bias e varianza.

**2. k-Fold Cross Validation**

Una variante che suddivide il dataset in **k sottoinsiemi (fold)** di dimensioni uguali, con ciascun fold usato una volta come validation set.

**Procedura:**

1. Suddividere casualmente i dati in kk fold.
2. Addestrare il modello su k−1k-1 fold e validarlo sul rimanente.
3. Ripetere per tutti i fold (kk iterazioni).
4. Calcolare le prestazioni medie su tutti i fold per valutare la scelta dei parametri.

**Vantaggi:**

* **Riuso dei dati**: tutti gli esempi sono usati sia per il training che per la validazione.
* **Riduzione della varianza**: metriche medie affidabili su più partizioni.

**3. Regolarizzazione**

La regolarizzazione controlla la complessità del modello aggiungendo un termine di penalizzazione alla funzione di errore.

**Punti chiave:**

1. **Formula**:

Errore totale=Errore sui dati+λ⋅Termine di penalizzazione\text{Errore totale} = \text{Errore sui dati} + \lambda \cdot \text{Termine di penalizzazione}

* + λ\lambda: controlla la forza della penalizzazione.
  + Penalizza modelli troppo complessi.

1. **Tipi di regolarizzazione**:
   * **Ridge Regression (L2)**: penalizza la somma dei quadrati dei parametri.
   * **Lasso Regression (L1)**: penalizza i valori assoluti, favorendo la selezione delle feature.
2. **Vantaggi**:
   * Riduce il rischio di overfitting.
   * Favorisce la generalizzazione.

**4. Tecniche per prevenire il sovradattamento**

1. **Regolarizzazione**: aggiunta di un termine di penalizzazione.
2. **Cross Validation**: testare il modello su diverse partizioni dei dati.
3. **Early Stopping**: interrompere il training quando le prestazioni sul validation set peggiorano.
4. **Riduzione della complessità del modello**:
   * Limitare profondità degli alberi o parametri.
5. **Aggiunta di dati** o **Data Augmentation**:
   * Generare nuovi dati per migliorare la generalizzazione.
6. **Riduzione delle feature**:
   * Selezionare solo le più rilevanti.

**5. Metodi basati su Kernel**

Utilizzati per affrontare problemi non lineari, proiettano i dati in spazi ad alta dimensione.

**Tipi di kernel:**

1. **Lineare**: adatto a dati separabili linearmente.
2. **Polinomiale**: cattura relazioni non lineari.
3. **RBF (Gaussian)**: misura la somiglianza tra punti.
4. **Sigmoid**: simula l'attivazione delle reti neurali.

**Applicazioni:**

* **SVM (Support Vector Machines)**:
  + Separano le classi massimizzando il margine.
  + L'iperpiano è calcolato nello spazio trasformato dal kernel.

**6. Ensemble Learning**

Unisce le predizioni di più modelli per migliorare le prestazioni.

**Tecniche principali:**

1. **Bagging**:
   * Riduce la varianza combinando modelli addestrati su sottoinsiemi casuali.
   * Esempio: **Random Forest**.
2. **Boosting**:
   * Riduce il bias addestrando modelli sequenzialmente per correggere errori.
   * Esempio: **Gradient Boosting**.
3. **Stacking**:
   * Combina modelli diversi con un meta-modello.

**7. Random Forest**

Un esempio di bagging applicato agli alberi di decisione:

* Ogni albero è addestrato su un sottoinsieme casuale di dati.
* Le predizioni finali sono una media (regressione) o un voto (classificazione).

**8. Functional Gradient Boosting**

Addestra modelli sequenzialmente per correggere errori residui:

1. Ogni iterazione aggiorna la predizione aggiungendo un contributo.
2. Minimizza la funzione di perdita iterativamente.

**9. Confronto tra tecniche di ensemble**

| **Tecnica** | **Obiettivo** | **Addestramento** | **Modelli base** | **Finalizzazione** |
| --- | --- | --- | --- | --- |
| Bagging | Ridurre varianza | Parallelamente | Indipendenti | Media o voto |
| Boosting | Ridurre bias | Incrementale | Sequenziali (dipendenti) | Combinazione ponderata |
| Stacking | Combinare modelli | Parallelamente | Indipendenti | Meta-modello |

**Quando usare:**

* **Bagging**: Modelli instabili, dataset ampi.
* **Boosting**: Problemi complessi, alto bias.
* **Stacking**: Modelli diversi, risorse computazionali elevate.