Model Ensemble - Bagging and Boosting

Hanno **due cose in comune**:

* Costruiscono **modelli predittivi multipli e diversi** a partire da versioni adattate di training data.
* **Combinano le previsioni** con una media o una votazione.

**Bagging**

Crea modelli diversi su diversi campioni casuali dell’insieme di dati originali. Campioni chiamati **bootstrap samples**. Le differenze tra i campioni creeranno diversità tra i modelli.

**Sample with Replacement**: Estrazione casuale dal training set e

* Ogni elemento può essere ri-estratto in un altro campione,
* Ogni campione ha la stessa DIM del training set originale,
* Ogni samples può lasciare fuori ⅓ dei data points (Out of Bag sample).

Per **combinare le previsioni** dei modelli si può usare:

* **Votazione**, vince la classe di maggioranza,
* **Media**, nel caso di regressione, si fa una media dei risultati

L’ensemble ha un confine decisionale che non può essere appreso da un singolo classificatore di base.

Bagging è **utile con modelli ad albero**, i quali sono sensibili alle variazioni dei dati di training. Consente di **ottenere una bassa varianza** della funzione di predizione stimata.

Quando applicato ai modelli ad albero, è combinato con **subspace sampling**, ogni albero viene costruito da un diverso sottoinsieme di feature. L’ensemble è detto **Random Forest**.

Nelle RF, date F feature, vengono selezionate f << F feature casuali.

**Random Forest**: Creano T bootstrap samples, Selezionano f feature random, Costruzione del decision tree per ogni sample, e poi si combinano le previsioni.

Se c’è un predittore molto forte, tutti gli alberi selezionano quello come root tree, e verranno simili. Per la classificazione si usa f = F1/2 con minimo 1, per la regressione f = ⅓ F con minimo 5.

**Out of Bag Error Estimation**

Tecnica di validazione per misurare l’errore di previsione. Gli esempi che non compaiono in ogni sample sono esempi inutilizzati. Permettono di convalidare, senza usare un test set separato.

Per ogni istanza del dataset originale si trovano gli alberi che non hanno incluso quella istanza nel loro bootstrap e si usano questi alberi per predire l’esito. Si raccolgono le previsioni per ogni istanza (usando voting o averaging). Confronto delle previsioni OOB con i valori effettivi per calcolare l’errore.

* Nella classificazione, percentuale di istanze classificate male,
* Regressione, MSE o MAE.

Entrambi i casi, valori bassi → prestazioni migliori.

**Boosting**

Idea: Formare classificatori in sequenza, e ognuno deve essere debole.

* Usa voting o averaging per combinare i risultati
* Combina modelli dello stesso tipo
* La ponderazione è usata per dare più influenza ai modelli più performanti
* È iterativo. Ogni nuovo modello è influenzato dalle prestazioni dei precedenti.
* Incoraggia i nuovi modelli a concentrarsi sulle istanze gestite in modo errato dai precedenti.

Al termine di ogni iter, il peso delle istanze corrette cala, mentre quello delle istanze classificate male aumenta. Alla iter successiva, si concentrerà sulle istanze difficili. Problema: Alcune difficili possono diventare più difficili e quelle facili, più facili e in generale tutte le possibilità possono verificarsi.

Tre motivi per cui può sbagliare a classificare:

* Istanze con valori uguali, ma classe diversa
* Mancanza di espressività. Esempio se i dati non sono linearmente separabili
* Varianza, è alta, se il confine decisionale dipende dai dati di training

Bagging riduce la varianza. Boosting riduce il bias.