Heterogenous Parallel Ensemble

08 December 2024 17:06

So far we've spoken of how ensembles combine multiple models to improve predictive accuracy, reduce bias + variance, all that.

Homogenous ensembles combine models of the same type.
Heterogenous ensembles combine models of diff types.

Multiple diff algos trained in $\|^{el}$ to tackle same task
→ Leverage strengths
→ Reduce individual model weakness.
↓
Better generalization + adaptability
Improved stability

Key combination techniques:
a. Weighting
b. Meta learning.

Weighting Techniques

Different weights to model on the basis of performance + reliability

Static

- fixed weights based on prior performance (proportional to accuracy on validation set)
- Final prediction = $\sum_{i=1}^{N} w_i \times model_i\ pred.$
- Pros:
  → simplicity
  → low computational cost
  → Consistency.
- Cons:
  → Inflexibility
  → Risk of underperformance

Dynamic

- Varying weights based on context of data. ('focus' on models that perform better for specific I/P)
- Pred. for (x) = $\sum w_i(x) \times model_i\ pred(x)$
- Pros:
  → Adaptability
  → Improved performance
  → Flexibility
- Cons:
  → Increased computational cost
  → More complex
  → Overfitting.

Adaptive weights:
→ I/P data characteristics
→ Model confidence
→ Contextual clues.

Meta Learning

- Also: stacking

meta learner learns optimal ways to combine preds from base models.

How?

Train base models ← 1. Train models separately on training dataset.
Gen. meta features ← 2. Use preds as I/P for meta learner.
Train meta learner ← 3. Train meta learner to predict final o/p.

gen. meta features ← 2. Use preds as I/P for meta learner.
Train meta learner ← 3. Train $\boxed{\text{meta learner}}$ to predict final o/p.

↓
Can be:
a. Log reg (binary classificⁿ)
b. Decision tree (classifⁿ + regression)
c. Neural network (high dimensional o/p)

- Adv:
  → captures non-linear relationships
  → Reduces overfitting

- Disadv
  → high computational cost.
  → May suffer data leakage

Feature	Weighting	Meta-Learning
Complexity	Lower	Higher
Flexibility	Moderate	High
Computational Cost	Moderate	High
Adaptability	Limited in static, moderate in dynamic	Very high
Best Use Case	When model reliability is known	When patterns among models are complex