Ensembles: Combine predictions from multiple models

Main Goal: Reduce variance

Secondary Goal (sometimes): Reduce bias

Example: Majority vote.

Suppose I have 3 classification models, 2 possible classes.
Each makes correct predictions for test set with probability 0.7, independently.

The majority vote classifier makes whatever prediction is made by at least 2 of the classifiers.

What is the probability the majority vote classifier is correct?

Define the following events:

A : first classifier is correct. $P(A) = 0.7$

A^c : first classifier incorrect. $P(A^c) = 0.3$

B : second classifier correct $P(B) = 0.7$

C : third classifier correct $P(C) = 0.7$

P(Majority vote classifier correct) = P(at least 2 correct)

$= P(A \text{ and } B \text{ and } C^c) + P(A \text{ and } B^c \text{ and } C)$
$+ P(A^c \text{ and } B \text{ and } C) + P(A \text{ and } B \text{ and } C)$

$= 0.7 \cdot 0.7 \cdot 0.3 + 0.7 \cdot 0.3 \cdot 0.7 + 0.3 \cdot 0.7 \cdot 0.7 + 0.7 \cdot 0.7 \cdot 0.7$

$= 0.784$

So the majority vote classifier does better than any individual classifier.

Caveat: Good classification methods are usually correlated.

What if the 3 classifiers all make identical predictions?

↳ total lack of independence.

- either all 3 are correct, or all 3 are incorrect
- Majority vote classifier does the same thing as any of the individual models. No improvement from ensemble.

$\boxed{\text{Ensembles are most effective when the baseline models you're combining are \underline{uncorrelated}}}$

One more comment on majority vote:

if you want class membership probabilities, take average:

$$\hat{f}_1^{(MV)}(x) = \frac{1}{3}\left(\hat{f}_1^{(1)}(x) + \hat{f}_1^{(2)}(x) + \hat{f}_1^{(3)}(x)\right)$$

~~Simple~~ Ensembles for Regression

$\hat{y}_i$ is a number.

Consider an ensemble that takes the average of predictions from 3 ~~stage 1~~ regression models:

$$\hat{y}_i^{(ensemble)} = \frac{1}{3}\left(\hat{y}_i^{(1)} + \hat{y}_i^{(2)} + \hat{y}_i^{(3)}\right)$$

Suppose component models are independent, have bias 0, and have the same variance of predicted values, σ^2 } Expected test set MSE $0^2 + \sigma^2 + Var(\varepsilon)$

- Bias 0: On average, $\hat{y}_i^{(1)} = f(x_i)$, $\hat{y}_i^{(2)} = f(x_i)$, $\hat{y}_i^{(3)} = f(x_i)$

$$E[\hat{y}_i^{(1)}] = E[\hat{y}_i^{(2)}] = E[\hat{y}_i^{(3)}] = f(x_i)$$

Therefore $E[\hat{y}_i^{(ensemble)}] = E\left[\frac{1}{3}\left(\hat{y}_i^{(1)} + \hat{y}_i^{(2)} + \hat{y}_i^{(3)}\right)\right]$

$$= \frac{1}{3}\left(E[\hat{y}_i^{(1)}] + E[\hat{y}_i^{(2)}] + E[\hat{y}_i^{(3)}]\right)$$

$$= \frac{1}{3} \cdot 3 \cdot f(x_i) = f(x_i)$$

So $\hat{y}_i^{(ensemble)}$ also has bias 0.

- Variance:

$$Var[\hat{y}_i^{(ensemble)}] = Var\left[\frac{1}{3}\left(\hat{y}_i^{(1)} + \hat{y}_i^{(2)} + \hat{y}_i^{(3)}\right)\right]$$

$$= \frac{1}{9}\left[Var(\hat{y}_i^{(1)}) + Var(\hat{y}_i^{(2)}) + Var(\hat{y}_i^{(3)})\right]$$

$$= \frac{1}{3}\sigma^2$$

- Expected test set MSE $0^2 + \underbrace{\tfrac{1}{3}\sigma^2}_{\text{reduced variance!}} + Var(\varepsilon)$
- Doesn't help if model predictions perfectly correlated!

Stacking: Fit a model that takes predictions from "component models" as input, predicts response.

- Basic motivation: Some models are better than others, we should give them more weight.

$$\hat{y}_i^{(ensemble)} = w_1 \hat{y}_i^{(1)} + w_2 \hat{y}_i^{(2)} + w_3 \hat{y}_i^{(3)}$$
$$= \beta_0^{\to 0} + \beta_1 \hat{y}_i^{(1)} + \beta_2 \hat{y}_i^{(2)} + \beta_3 \hat{y}_i^{(3)}$$

w_1 is weight assigned to model 1, large if model is good & small if model 1 is bad.

we could optionally impose the constraint that

$0 \le w_m \le 1$ and $\sum_m w_m = 1$.

- We can estimate model weights based on training set performance
- We should get weights based on cross-validated performance, or else we will give too much weight to models that overfit training data.

Process:

Estimation:

1. Get cross-validated predictions for each "stage 1" or "component" model
2. Create a new data set where the "explanatory variables" (design matrix X) are the cross-validated predictions from the component models
3. Fit a "stage 2"/ensemble model to predict the response based on component model predictions

Prediction:

4. Re-fit each component model to the full training set & get predictions for the test set
5. Create a new data set with test set predictions from component models
6. Predict using stage 2 model from step 3 & predictions from step 5.