Committees of Models

Sunday, April 14, 2013 10:24 PM

random forest models

multiple CART models fitted on bootstrapped data sets drawn from the sample

example of "bagging" — bootstrap aggregating

Bishop: "Committees" of model

committee's prediction $y_{com}(x)$

$$= \frac{1}{M} \sum_{m=1}^{M} y_m(x)$$

where m indexes each model

M = # of models

$h(x)$: DGP $\qquad y_m(x) = h(x) + \varepsilon_m(x)$

$$E_x[SSE_m] = E_x\left[\left(y_m(x) - h(x)\right)^2\right] = \underbrace{E_x\left[\varepsilon_m(x)^2\right]}$$

the expectation here is over x.

For the entire committee,

$$E[SSE_{com}] = E\left[\left(\frac{1}{M}\sum_{m=1}^{M} y_m(x) - h(x)\right)^2\right] = \overline{E\left[\left(\frac{1}{M}\sum_{m=1}^{M} \varepsilon_m(x)\right)^2\right]}$$

start omitting dependency of ε on x.

$$E\left[\frac{1}{M^2}\cdot(\varepsilon_1+\varepsilon_2+\cdots+\varepsilon_M)(\varepsilon_1+\varepsilon_2+\cdots\varepsilon_M)\right]$$

If we assume $\mu_{\varepsilon_m}=0$ and $\varepsilon_m, \varepsilon_k$ are statistically independent for all $m, k \in 1\ldots M$ then

$$\frac{1}{M^2}\ E\left[\left(\sum_{m=1}^{M}\varepsilon_m-\mu_{\Sigma\varepsilon}\right)\cdot\left(\sum_{m=1}^{M}\varepsilon_m-\mu_{\Sigma\varepsilon}\right)\right]$$

$$\mu_{\Sigma\varepsilon}=\frac{1}{M}\cdot E\left[\varepsilon_1+\varepsilon_2+\cdots+\varepsilon_M\right]$$

by independence,

$$\frac{1}{M}\cdot E[\varepsilon_1]+E[\varepsilon_2]+\cdots+E[\varepsilon_M]$$

by assumption, $E[\varepsilon_m]=0 \quad \forall\, m\in 1..M$

ergo,

$$\mu_{\Sigma\varepsilon}=0$$

$$\frac{1}{M^2}\cdot \operatorname{var}\left[\sum_{m=1}^{M}\varepsilon_m\right]=\frac{1}{M^2}\cdot\sum_{m=1}^{M}\operatorname{var}(\varepsilon_m)$$

b/c εs are statistically independent.

so, because $\mu_m=0 \quad \forall\, \varepsilon_m,\ m\in 1\ldots M$,

$$\operatorname{var}(\varepsilon_m)=E\left[\varepsilon_m^2\right]$$

putting dependency on x back in.

$$E[SSE_{com}]=\frac{1}{M^2}\sum_{m=1}^{M}E\left[\varepsilon_m(x)^2\right]$$

?

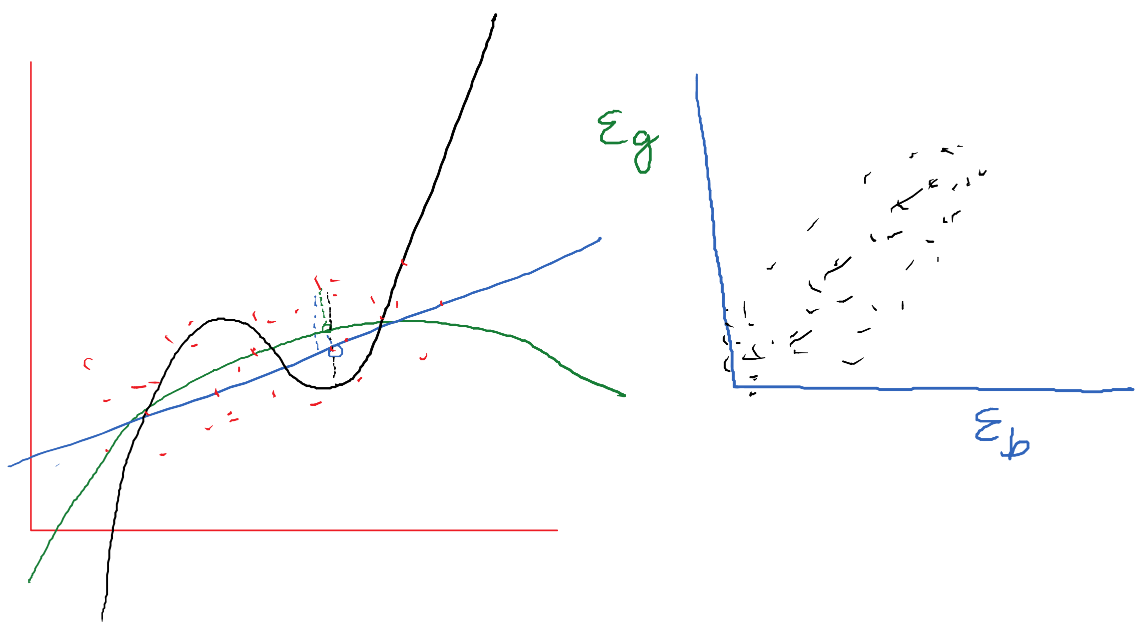
$$E\left[SSE_M\right] = E_x\left[\varepsilon_M(x)^2\right] \quad ?$$

well, $E\left[SSE_{COM}\right] = \frac{1}{M^{\cancel{2}}} \cdot \cancel{M}\, E\left[SSE_M\right]$

$$E\left[SSE_{COM}\right] = \frac{1}{M} E\left[SSE_M\right]$$

the expected SSE of an ensemble estimator is equal to $\frac{1}{M}$ times the expected SSE of its M many constituents . . .

. . . as long as ε_M are statistically independent.



it's hard to create models of the same data with independent $\varepsilon_M(x)$.

under perfect error correlation,

$$E[SSE_{com}] = E\left[\frac{1}{M^2}\cdot(\varepsilon_1+\varepsilon_2+\cdots+\varepsilon_M)(\varepsilon_1+\varepsilon_2+\cdots\varepsilon_M)\right]$$

$$= E\left[\frac{1}{\cancel{M^2}}\cdot \cancel{M}\cdot\varepsilon_M\cdot\cancel{M}\cdot\varepsilon_M\right]$$

$$= E\left[\varepsilon_M^2\right] = E\left[SSE_M\right].$$

... the same as any constituent model. Ergo there is no benefit to constructing an ensemble.

lower bound $E[SSE_{com}] = \frac{1}{M}E[SSE_M]$

upper bound $E[SSE_{com}] = E[SSE_M]$

... in other words, ensemble models can't do WORSE than their (average) constituent, but can do much better.

Boosting

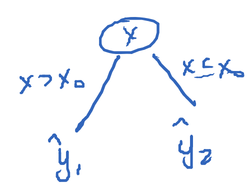
Sunday, April 14, 2013 10:24 PM

fitting a series of models sequentially to a data set, where each model improves upon the last by focusing upon the observations for which the previous model was most deficient.

Adaboost ("adaptive boosted") is one implementation of boosting. Weights misclassified/poorly predicted cases more heavily when fitting subsequent models.

Adaboost

"decision stump"



(1) set $w_n^{(1)} = \frac{1}{N}$ for all observations $n = 1 \ldots N$.

This is the observation weighting.

(2) start with your first classifier.

Fit this classifier to minimize:

$$\sum_{n=1}^{N} \underbrace{w_n^{(m)}}_{\text{indicator weight for classifier } m.} \cdot \underbrace{I\Big(\overbrace{y_m(x_n)}^{\text{classifier m's prediction for } y_n \text{ given } x_n \text{ (IVs for obs n)}} \neq \boxed{t_n}\Big)}_{\text{indicator: 1 if true, 0 if false}}$$

t_n → target dv for obs n

in words: minimize weighted classification error.

③ Calculate

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} \cdot I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

if for m=1 we misclassified EVERY obs,

$$\varepsilon_m = \frac{\sum_{n=1}^{N} \frac{1}{N} \cdot 1}{\frac{1}{N} \cdot N} = \frac{\cancel{\frac{1}{N}} \cdot 1 \cdot \cancel{N}}{\cancel{\frac{1}{N}} \cdot \cancel{N}} = 1$$

and if we correctly classified EVERY case

$$\varepsilon_m = \frac{\sum \frac{1}{N} \cdot 0}{\sum \frac{1}{N}} = 0 .$$

④ update $w_n^{(m+1)} = w_n^{(m)} \cdot \exp\left(\alpha_m \cdot I(y_m(x_n) \neq t_n)\right)$

$$\alpha_m = \ln\left[\frac{1 - \varepsilon_m}{\varepsilon_m}\right]$$

all for $m=1$

under "expected awful" classifier (coin flip)

$$\varepsilon_m \approx \frac{\frac{1}{\cancel{N}} \cdot 1 \cdot \frac{1}{2} \cancel{N}}{\frac{1}{\cancel{N}} \cdot \cancel{N}} \approx \frac{1}{2}.$$

$$\alpha_m = \ln\left(\frac{1 - 1/2}{1/2}\right) = \log(1) = 0.$$

$\longrightarrow$ no weight updating.

if we classify 3/4 correct:

$$\varepsilon_m \approx \frac{\frac{1}{N} \cdot 1 \cdot \frac{1}{4} N}{\frac{1}{N} \cdot N} = \frac{3}{4}.$$

$$\alpha_m = \ln\left(\frac{1 - 1/4}{1/2}\right) = \ln\left(\frac{3/4}{1/2}\right)$$

$$= \ln\left(\frac{3}{2}\right) \approx 0.40.$$

$\longrightarrow$ $w_n^{m+1} = \frac{3}{2} \cdot w_n^m$ IF n was misclassified.

$w_n^{m+1} = w_n^m$ IF n was correctly classified.

(5) Repeat 2-4 for $m = 1 \dots M$.

(6) Calculate final predictions:

$$y_M(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m \cdot y_m(x)\right]$$

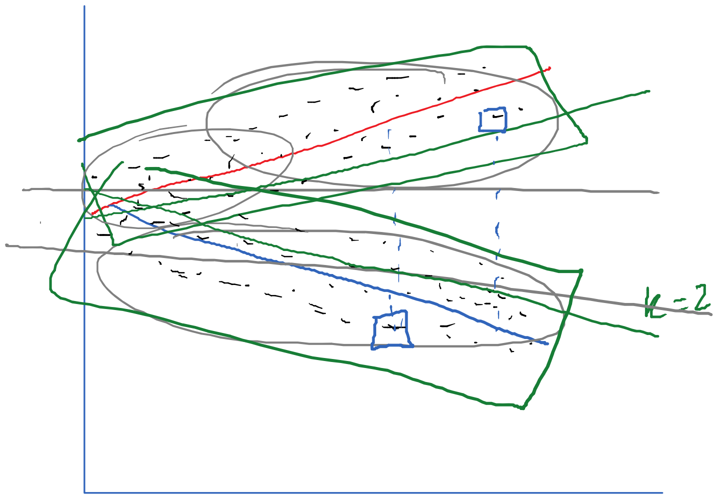
$$y_m \in \{-1, 1\}$$

Adaboost: designed for discrete DVs.

For continuous DVs, "gradient boosting" is a similar algorithm; each classifier is fit to residual errors $t_n - f_{(m-1)}(x)$

Conditional Mixture Models

Sunday, April 14, 2013 10:30 PM



K = # of distinct models

Conditional regression mixture model

$$\ln L = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \cdot \ln\left(\pi_k \, \Phi\left(t_n \mid x\beta_k, \sigma_k^2\right)\right)$$

z_{nk}: latent parameter $\{0,1\}$ indicating whether obs. n belongs to model k.

π_k: probability that any given data point is in Model k

$\Phi(t_n \mid x\beta_k, \sigma_k^2)$: likelihood of t_n given model k's slope, int., and σ_k^2 parameters.

Maximization is via the EM algorithm

E-step: $\gamma_{nk} = E[z_{nk}] = \dfrac{\pi_k \, \Phi(t_n \mid x_n \beta_k, \sigma_k^2)}{\quad}$

$$\sum_j \pi_j \, \Phi(t_n \mid X_n \beta_j, \sigma_j^2)$$

M-step: $\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}$

regression step: WLS for each model k, weights given by $\underset{n \times n}{\mathrm{diag}}(\gamma_{nk}) = R$

$$\beta^* = (X' R X)^{-1} X' R t$$

matrix of IVs including a constant.

target DV

Bayesian Model Averaging

Sunday, April 14, 2013 10:31 PM

BMA is directed at averaging predictions over a variety of models on the same data set, with the idea that we are not certain which of the models is "true" and we want to incorporate this uncertainty into our predictions.

Ensemble: $\int y(x, z) f(z)\, dz = y(x)$

BMA: $\int y_m(x, z)\, f(m)\, dm = y(x)$

$y \sim \quad x, z, w, q, r$

Not sure about specification.

$y \sim z$

$y \sim x + z$

$y \sim x + w$

$y \sim x + z + w$

$y \sim \beta_0 + \beta_1 z + \dots$

$$\breve{y} \sim x + z + w$$
$\vdots$

$$y \sim \beta_0 + \underline{\beta_1 z} + \cdots$$

Run many models, and then combine results.

E.g. for $\hat{\beta}_z = \sum_i w^{(s)} \hat{\beta}_z^{(s)}$ for all candidate models s

but then $w^{(s)}$ is set according to the "likelihood" (posterior probability) the model is the best one.

BIC weights: $$w^{(s)} = \frac{\exp(\frac{1}{2} BIC_s^*)}{\sum_s \exp(\frac{1}{2} BIC_s)}$$

(assuming BIC is positively associated w/ model quality)

$$BIC_s = 2 \ln L_s - \log n \cdot k_m$$

normalize BIC: $$BIC_s - \max_t (BIC_t) = BIC_s^*$$

"leaps and bounds" algorithm $\ast$

"Occam's window"

1 only select models:

quality

$$\frac{\max_t (BIC_t)}{BIC_s} \leq C = 20$$

2 delete a model (t) if

parsimony

$$\frac{BIC_s}{BIC_t} > 1 \quad \text{and} \quad s \subset t$$

in words: if t contains everything in s but has a lower BIC .. exclude it.