Tuesday, 10/27/2020

EE 660

MACHINE LEARNING FROM SIGNALS: FOUNDATIONS AND METHODS

Prof. B. Keith Jenkins

Lecture 19

p.1

Lecture 19 **EE 660** **Oct 27, 2020**

Announcements

- Homework 7 is due Friday
- Project proposals are being graded

Reading

- Boosting: Murphy 16.4.0 - 16.4.4

Today's topics

- Variance of an average
- Random Forest
- Boosting (part 1)

Variance of an Average

$$(1) \quad \text{var}(\underline{x}) = \mathbb{E}_{\mathcal{D}}\left\{\left(\underbrace{h_g^{(\mathcal{D})}(\underline{x})} - \bar{h}_g(\underline{x})\right)^2\right\}$$

Take average h_g over many draws of a dataset $\mathcal{D}_b$ from $\mathcal{D}$ (drawn with replacement)

Each $\mathcal{D}_b$ will be used to train a tree → $h_g^{(\mathcal{D}_b)}(\underline{x})$.

Let $\tilde{h}_g^{(\mathcal{D})}(\underline{x}) \triangleq \frac{1}{B}\sum_{b=1}^{B} h_g^{(\mathcal{D}_b)}(\underline{x})$

Then:

$$(1') \quad \text{var}_{\text{ave}}(\underline{x}) = \mathbb{E}_{\mathcal{D}}\left\{\left(\tilde{h}_g^{(\mathcal{D})}(\underline{x}) - \bar{h}_g(\underline{x})\right)^2\right\}$$

How does $\text{var}_{\text{ave}}(\underline{x})$ compare with $\text{var}(\underline{x})$?

If we take average of B i.i.d. random variables v_i, $i=1,2,\cdots,B$, each with variance σ^2, the average $v_{ave} = \frac{1}{B}\sum_{i=1}^{B} v_i$ will have variance:

$$\sigma_{ave}^2 = \frac{\sigma^2}{B}$$

If instead, the r.v. v_i are identically distributed but have positive pairwise correlation ρ, one can show that:

(2)

$$\boxed{\sigma_{ave}^2 = \rho\sigma^2 + (1-\rho)\frac{\sigma^2}{B}, \qquad 0 \le \rho \le 1}$$

in which ρ = correlation coefficient:

$$\rho \triangleq \frac{\mathbb{E}\{v_i v_j\} - \mathbb{E}\{v_i\}\mathbb{E}\{v_j\}}{\sigma_{v_i}\sigma_{v_j}} = \frac{\mathbb{E}\{v_i v_j\} - \mu_v^2}{\sigma_v^2}$$

$\therefore$ Taking average over a set of trees to get $\tilde{h}_g^{(B)}(\underline{x})$, can reduce the variance — less pairwise correlation between trees gives more reduction in variance.

Random Forests

(a) Draw many datasets $\mathcal{D}_b$ from $\mathcal{D}_{Tr}$, with replacement

→ each $\mathcal{D}_b$ gives rise to a tree T and est. $\hat{f}_b(\underline{x})$.

At each point $\underline{x}$,
take average result $\hat{f}(\underline{x}) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(\underline{x})$ (regression)
or take vote (classification)

→ this by itself is called <u>bagging</u> (for "bootstrap aggregating")

⇒ Datasets are (highly) correlated (esp. if $|\mathcal{D}_b| \approx |\mathcal{D}_{Tr}|$)

⇒ resulting $\hat{f}_b(\underline{x})$ are (highly) correlated

⇒ variance reduces some, but not a lot.

(b) Before splitting each region R_m,
- select a random subset of d features ($d < D$ or $d \ll D$)
- then select best feature out of the subset to threshold

⇒ correlation between trees is typically much smaller

⇒ variance reduces by a lot more.

Algorithm – Random Forest [Hastie, et al., Algorithm 15.1]

```
1. For $b=1$ to $B$

   (a) Draw a sample dataset $\mathcal{D}^*$ at random, with replacement, of size $N^*$,
       from $\mathcal{D}_{Tr}$. Typically, $N^* = N_{Tr}$.

   (b) Grow a random-forest tree $T$, using $\mathcal{D}^*$:
       Cycle through each region $R_m$; for each $R_m$:

           (i) Select $d$ features at random (from all $D$ features) [1]

           (ii) Use CART method to split $R_m$ by finding optimal $j$, $t_k$, $\underline{w}$

           (iii) Split the tree node into 2 daughter nodes

       Iterate until a halting condition is reached (see CART conditions)

2. Output the set of trees $\{T_b,\ b=1,2,\dots,B\}$
```

[1] Common choices: $d = \lfloor \sqrt{D} \rfloor$ (classification); $d = D/3$ (regression). Best to adjust & choose using model selection.

3. Use $\{T_b\}$ for prediction:

Regression:

$$\hat{f}(\underline{x}) = \frac{1}{B}\sum_{b=1}^{B} \underbrace{\hat{f}_b(\underline{x})}_{\text{prediction from tree } T_b(\underline{x})}$$

$$= \frac{1}{B}\sum_{b=1}^{B} \underbrace{\sum_{m=1}^{M_b} w_m^{(b)}\, \mathbb{I}(\underline{x} \in R_m^{(b)})}_{\text{prediction from tree } T_b(\underline{x})}$$

$$= \frac{1}{B}\sum_{b=1}^{B} \sum_{m=1}^{M_b} w_m^{(b)}\, \phi(\underline{x}; \underline{v}_m^{(b)})$$

→ Resulting $\hat{f}(\underline{x})$ will still be a piecewise constant fcn. of $\underline{x}$.

Classification:

Class prediction: Let $\hat{y}^{(b)}(\underline{x})$ be class assignment from tree T_b

$$\hat{y}(\underline{x}) = \operatorname*{argmax}_{c} \sum_{b=1}^{B} \mathbb{I}\left[\hat{y}^{(b)}(\underline{x}) = c\right]$$

($\hat{y}$ is class label with most predictions among B trees)

Can also estimate class posterior probabilities $p(\hat{y}=c \mid \underline{x}, \mathcal{D})$, by:

Let $p_c^{(b)}(\underline{x})=$ freq. of occurence of data pts. $y_i = c$ in $R_m^{(b)}$ that contains $\underline{x}$, from tree T_b.

At each pt. $\underline{x}$, take average:

$$p(\hat{y}=c \mid \underline{x}, \mathcal{D}) \approx \frac{1}{B} \sum_{b=1}^{B} p_c^{(b)}(\underline{x}).$$

→ R.F. tends to perform much better than single-tree CART.

[Fig. 15.1 of Hastie, et. al]