LaTeX-Stats-Template

Berent Å. S. Lunde   Jens Christian Wahl

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Workshop or Event
Place, Location
15th February 2021

Agenda

1. Blocks
2. Algorithms
3. Section3Title

Custom blocks

Blocks can have different colours for effect.

Poisson GLM might look good with yellow block

- $N_i \sim \text{Poisson}(\lambda_i)$ for observation i.

$$\log \lambda_i = \mathbf{x}_i^T \beta \text{ (GLM)}$$

$$\log \lambda_i = \mathbf{x}_i^T \beta + \sum_j f_j(x_{ij}) \text{ (GAM)}$$

Gamma GLM might loog good inside the green block

- $S_i \sim \text{Gamma}(\mu_i, \nu)$ where $\mathbb{E}(S_i) = \mu_i$.

$$\log \mu_i = \mathbf{x}_i^T \beta \text{ (GLM)}$$

$$\log \mu_i = \mathbf{x}_i^T \beta + \sum_j f_j(x_{ij}) \text{ (GAM)}$$

Claim frequency and claim severity

Model for claim frequency

Assume $N_i \sim \text{Poisson}(\lambda_i)$ for customer i.

$$\log \lambda_i = \boldsymbol{x}_i^T \boldsymbol{\beta} \quad \text{(GLM)}$$

$$\log \lambda_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \sum_j f_j(x_{ij}) \quad \text{(GAM)}$$

Model for claim size

Assume $S_i \sim \text{Gamma}(\psi_i, \nu)$ where $\mathbb{E}(S_i) = \psi_i$.

$$\log \psi_i = \boldsymbol{x}_i^T \boldsymbol{\beta} \quad \text{(GLM)}$$

$$\log \psi_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \sum_j f_j(x_{ij}) \quad \text{(GAM)}$$

Split frame and visible

Left split col

- Make good points

Split frame and visible

Left split col

- Make good points
- Sequentially

Split frame and visible

Left split col

- Make good points
- Sequentially
- For impact!

Split frame and visible

Left split col

- Make good points
- Sequentially
- For impact!

Right split col

- Show this column later
- Give more flexibility
- Write amazing things

Algorithm 1 AdaBoost: First boosting algorithm

Input:
- A training set $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$,
- a family of base-learners $\mathcal{H}$,

Do:
1. Initialize the training weights: $w_i = \frac{1}{n}, \ i = 1, \cdots, n$
2. **for** $k = 1$ to K:

   i) Fit a classifier, $f_k(\mathbf{x})$, to the training data using weights w_i.
   $$f_k = \arg\min \sum_{i=1}^n w_i I(y_i \neq f(\mathbf{x}_i))$$
   ii) Compute model-weight
   $$\alpha_k = \log\left(\frac{1-\mathrm{err}_k}{\mathrm{err}_k}\right), \ \mathrm{err}_k = \frac{\sum_{i=1}^n w_i I(y_i \neq f(\mathbf{x}_i))}{\sum_{i=1}^n w_i}$$
   iii) Recompute training weights:
   $$w_i \leftarrow w_i \exp\left(\alpha_k I(y_i \neq f_k(\mathbf{x}_i))\right), \ i = 1, \cdots, n$$
   end for
3. **Return** $f^{(K)}(\mathbf{x}) = \sum_{k=1}^K \alpha_k f_k(\mathbf{x})$ to use with `sign()`.

Comment here that the exponential loss is crucial for AdaBoost.

Algorithm environment counts

Algorithm 2 First order gradient boosting

Input:
- A training set $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$,
- a differentiable loss $l(y, f(\mathbf{x}))$,
- a family of base-learners $\mathcal{H}$,

Do:

1. Initialize model with a constant value:
   $f^{(0)} = \underset{\eta}{\arg\min} \sum_{i=1}^n l(y_i, \eta)$.
2. **for** $k = 1$ to K:
   i) Compute derivatives g_i for all $i = 1 : n$.
   ii) Fit a base-learner $f_k(\mathbf{x}) \in \mathcal{H}$ to $\{-g_i, \mathbf{x}_i\}_{i=1}^n$ using MSE-loss.
   iii) Find an optimized scaling α_k of f_k:
   $\hat{\alpha}_k = \underset{\alpha}{\arg\min} \sum_{i=1}^n l(y_i, f^{(k-1)}(\mathbf{x}_i) + \alpha f_k(\mathbf{x}_i))$.
   v) Update the model with a scaled base-learner (δ "small"): $f^{(k)}(\mathbf{x}) = f^{(k-1)}(\mathbf{x}) + \delta \hat{\alpha}_k f_k(\mathbf{x})$.
   end for
3. **Return** $f^{(K)}(\mathbf{x})$.

Blue and green may be used to signify something important. For example differences from Algorithm 1.

Specific R-coding

```r
x <- runif( 500, 0, 4 )
y <- rnorm( 500, x, 1 )
x.test <- runif( 500, 0, 4 )
y.test <- rnorm( 500, x.test, 1 )
# Train gbtorch ensemble
mod <- gbt.train( y, as.matrix(x) )
y.pred <- predict( mod, as.matrix( x.test ) )
# Plot predictions on test data
plot( x.test, y.test )
points( x.test, y.pred, col="red" )
```

Section outline

- Amazing things
- more amazing things.

Questions?