Loss Functions & Model Performance

Justin Post

Loss Functions

- Loss functions are the function that we use to **fit** or **train** our model

Loss Functions

- Loss functions are the function that we use to **fit** or **train** our model
- Ex: MLR with minimizing the sum of squared errors
  - Response: y = brozek score
  - Predictors: x_1 = age, x_2 = height, ...

$$\min_{\beta's} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}))^2$$

Loss Functions

- Loss functions are the function that we use to **fit** or **train** our model
- Ex: MLR with minimizing the sum of squared errors plus a penalty
  - Response: y = brozek score
  - Predictors: x_1 = age, x_2 = height, ...

$$\min_{\beta's}\sum_{i=1}^{n}(y_i-(\beta_0+\beta_1x_{1i}+\ldots+\beta_px_{pi}))^2+\alpha\sum_{j=1}^{p}|\beta_j|$$

NC STATE UNIVERSITY

Loss Functions

- Loss functions are the function that we use to **fit** or **train** our model

- Ex: MLR with minimizing the mean absolute error

  - Response: y = brozek score
  - Predictors: x_1 = age, x_2 = height, ...

$$\min_{\beta's} \sum_{i=1}^{n} |y_i - (\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi})|$$

Loss Functions

- Loss functions are the function that we use to **fit** or **train** our model
- Ex: Logistic Regression with (negative) binary cross entropy
  - Response: y = Potability (1 or 0)
  - Predictors: x_1 = Hardness, x_2 = Chloramines, ...

$$\min_{\beta's} = -\sum_{i=1}^{n}(y_i log(p(x_1,\ldots,x_n)) + (1-y_i)log(1-p(x_1,\ldots,x_n)))$$

where $p(x_1,\ldots,x_n) = \frac{1}{1+e^{-\beta_0-\beta_1 x_{1i}-\ldots-\beta_p x_{pi}}}$

NC STATE UNIVERSITY

Loss Functions

- Loss functions are the function that we use to **fit** or **train** our model
- Ex: Logistic Regression with (negative) binary cross entropy and penalty
    - Response: y = Potability (1 or 0)
    - Predictors: x_1 = Hardness, x_2 = Chloramines, ...

$$\min_{\beta' s} = -\sum_{i=1}^{n}(y_i log(p(x_1,\ldots,x_n)) + (1-y_i)log(1-p(x_1,\ldots,x_n))) + \lambda\sum_{i=1}^{p}\beta_j^2$$

where $p(x_1,\ldots,x_n) = \frac{1}{1+e^{-\beta_0-\beta_1 x_{1i}-\ldots-\beta_p x_{pi}}}$

NC STATE UNIVERSITY

Model Metric

- Model metrics are used to determine the quality of the predictions
- Pretty much any loss function can also act as a metric!
  - Often choose to use the same loss function used as the metric

Model Metric

- Model metrics are used to determine the quality of the predictions
- Pretty much any loss function can also act as a metric!
    - Often choose to use the same loss function used as the metric
- Ex:
    - Fit 'usual' least squares regression (minimize sum of squared errors)
    - Determine quality with RMSE or mean absolute error (MAE)

Model Metric

- Model metrics are used to determine the quality of the predictions
- Pretty much any loss function can also act as a metric!
    - Often choose to use the same loss function used as the metric
- Ex:
    - Fit (MLR) LASSO model (minimize sum of squared errors subject to L1 penalty)
    - Determine quality with RMSE or MAE

NC STATE UNIVERSITY

Model Metric

- Model metrics are used to determine the quality of the predictions
- Pretty much any loss function can also act as a metric!
    - Often choose to use the same loss function used as the metric
- Ex:
    - Fit Logistic Regression model (minimize (negative) binary cross entropy)
    - Determine quality with (negative) binary cross entropy (`neg_log_loss`) or accuracy

Other Commonly Used Model Metrics

For a categorical response, many rely on:

- "Wolf" is a **positive class**.
- "No wolf" is a **negative class**.

We can summarize our "wolf-prediction" model using a 2x2 confusion matrix that depicts all four possible outcomes:

True Positive (TP): • Reality: A wolf threatened. • Shepherd said: "Wolf." • Outcome: Shepherd is a hero.	**False Positive (FP):** • Reality: No wolf threatened. • Shepherd said: "Wolf." • Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): • Reality: A wolf threatened. • Shepherd said: "No wolf." • Outcome: The wolf ate all the sheep.	**True Negative (TN):** • Reality: No wolf threatened. • Shepherd said: "No wolf." • Outcome: Everyone is fine.

From google's ML crash course

Other Commonly Used Model Metrics

For a categorical response:

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision = $\frac{TP}{TP+FP}$
- Recall (or True positive rate, TPR) = $\frac{TP}{TP+FN}$
- False Positive Rate (FPR) = $\frac{FP}{FP+TN}$

Other Commonly Used Model Metrics

For a categorical response:

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision = $\frac{TP}{TP+FP}$
- Recall (or True positive rate) = $\frac{TP}{TP+FN}$
- False Positive Rate (FPR) = $\frac{FP}{FP+TN}$

Built off of these ideas

- Receiver Operating Characteristic (ROC) curve
- Plots FPR vs TPR at different classification thresholds
- Area under ROC curve often used!

Note: Model Selection Without Training/Test

- For a numeric response, these are just calculated on the training data
    - AIC
    - AICc
    - BIC
    - Mallow's Cp
    - Adjusted R-squared
- Can be used to select a model without a training/test split

Recap

- Loss functions are used during model fitting
- Model metrics are used to evaluate a model
  - Can be the same!
  - Often still call it a loss function when using as a metric