Loss Functions

Prof. Asim Tewari
IIT Bombay

Mean Square Error/Quadratic Loss/L2 Loss

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

- Due to squaring, predictions which are far away from actual values are penalized heavily in comparison to less deviated predictions. Plus MSE has nice mathematical properties which makes it easier to calculate gradients.

Mean Absolute Error/L1 Loss

$$MAE = \frac{\sum_{i=1}^{n} | \, y_i - \hat{y}_i \, |}{n}$$

- Unlike MSE, MAE needs more complicated tools such as linear programming to compute the gradients. Plus MAE is more robust to outliers since it does not make use of square.

Mean Bias Error

$$MBE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}$$

- This is same as MSE with the only difference that we don't take absolute values. Clearly there's a need for caution as positive and negative errors could cancel each other out. Although less accurate in practice, it could determine if the model has positive bias or negative bias.

Loss + Penalty

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \left\{ L(\mathbf{X}, \mathbf{y}, \beta) + \lambda P(\beta) \right\}$$

Ridge and Lasso Loss

$$L(\mathbf{X}, \mathbf{y}, \beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

For ridge regression $P(\beta) = \sum_{j=1}^{p} \beta_j^2$

For lasso regression $P(\beta) = \sum_{j=1}^{p} |\beta_j|$

Classification Losses

In Classification the dataset is $\{(x_i, y_i)\}_{i=1}^{n}$, where

$x_i \in \Re^d$ and $y_i \in \{-1, 1\}$

And we need to build a classifier which reduces the loss.

What is loss is classification?

How close is your prediction to the actual value.

Classification Losses

What is loss is classification?

How close is your prediction to the actual value.

But the actual value is a class, either you predict it correctly or you do not, there is no other value between the binary answer.

Zero-one loss

$$L(y, \hat{y}) = \begin{cases} 1 & if\ y \neq \hat{y} \\ 0 & if\ y = \hat{y} \end{cases}$$

Classification Losses

Logistic loss

$\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \Re^d$ and $y_i \in \{0, 1\}$

$$L(y, \hat{y}) = -\{y ln(\hat{y}) + (1 - y)\ln(1 - \hat{y})\}$$

Classification Losses (Surprise function)

$\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \Re^d$ and $y_i \in \{1, 2, .. K\}$

Surprise function

The surprise of observing that a discrete random variable Y takes on value k is:

$$\log \frac{1}{P(Y=k)} = -\log P(Y=k)$$

As P(Y = k) → 0, the surprise of observing that value approaches ∞, and conversely as P(Y = k) → 1, the surprise of observing that value approaches 0.

The entropy of Y, denoted H(Y), is the expected surprise:

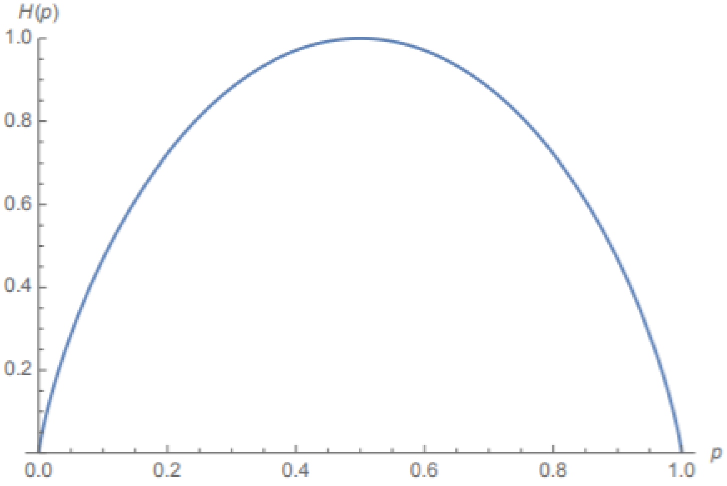
$$\begin{aligned} H(Y) &= \mathbb{E}[-\log P(Y)] \\ &= -\sum_k P(Y=k) \log P(Y=k) \end{aligned}$$

Classification Losses (Entropy)

$\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \Re^d$ and $y_i \in \{1, 2, .. K\}$

The entropy of Y, denoted H(Y), is the expected surprise:

$$\begin{aligned} H(Y) &= \mathbb{E}[-\log P(Y)] \\ &= -\sum_k P(Y = k) \log P(Y = k) \end{aligned}$$



A graph of entropy vs. p for a Bernoulli(p) random variable

Classification Losses (Entropy)

The entropy of Y, denoted H(Y), is the expected surprise:

$$\begin{aligned} H(Y) &= \mathbb{E}[-\log P(Y)] \\ &= -\sum_k P(Y=k) \log P(Y=k) \end{aligned}$$

$\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \Re^d$and $y_i \in \{1, 2, ..K\}$

This definition is for random variables, but in practice we work with data. The distribution is empirically defined by our training points. Concretely, the probability of class k is the proportion of datapoints having class k:

$$P(Y=k) = \frac{|\{i \mid y_i = k\}|}{n}$$

Classification Losses (Entropy)

We know that when we choose a split-feature, split-value pair, we want to reduce entropy in some way. Let $X_{j,v}$ be an indicator variable which is 1 when $x_j < v$, and 0 otherwise. There are a few entropies to consider:

$H(Y)$

$H(Y|X_{j,v} = 1)$, the entropy of the distribution of points such that $x_j < v$.

$H(Y|X_{j,v} = 0)$, the entropy of the distribution of points such that $x_j \geq v$.

We want to split the points in to two groups based on when $x_j < v$, such that we want to minimize a weighted average entropy of the two sides of the split, where the weights are proportional to the sizes of two sides:

$$\text{minimize } H(Y|X_{j,v}) := P(X_{j,v} = 1)H(Y|X_{j,v} = 1) + P(X_{j,v} = 0)H(Y|X_{j,v} = 0)$$

An equivalent way of seeing this is that we want to maximize the information we've learned, which is represented by how much entropy is reduced after learning whether or not $x_j < v$:

$$\text{maximize } I(X_{j,v}; Y) := H(Y) - H(Y|X_{j,v})$$

Classification Losses (Gini impurity)

Another way to assess the quality of a split is **Gini impurity**, which measures how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. It as defined as:

$$G(Y) = \sum_k P(Y=k) \sum_{j \neq k} P(Y=j) = \sum_k P(Y=k)(1 - P(Y=k)) = 1 - \sum_k P(Y=k)^2$$

Exactly as with entropy, we can define a version of this quantity which is dependent on the split. For example, *$G(Y \,|X_{j,v} = 1)$* would be the Gini impurity computed only on those points satisfying *$x_j < v$*. And we can define an analogous quantity

$$G(Y|X_{j,v}) := P(X_{j,v} = 1)G(Y|X_{j,v} = 1) + P(X_{j,v} = 0)G(Y|X_{j,v} = 0)$$

which is to be minimized.

Empirically, the Gini impurity is found to produce very similar results to entropy, and it is slightly faster to compute because we don't need to take logs.

Classification Losses (misclassification rate)

Since we ultimately care about classification accuracy, it is natural to wonder why we don't directly use the misclassification rate by plurality vote as the measure of impurity:

$$M(Y) = 1 - \max_k P(Y = k)$$

This is not very sensitive for evaluating the splits.

Classification Losses (misclassification rate)

$$M(Y) = 1 - \max_k P(Y = k)$$

This is not very sensitive for evaluating the splits.

Original data: 80 points
40 of class A
40 of class B

Classifier 1

Set 1(A): 30 of class A and 10 of class B
M(Y_1|Xj,v = 1) = 10/40 = 1/4
P(Xj,v = 1)=40/80 = 1/2

Set 2(B): 10 of class A and 30 of class B
M(Y_2| Xj,v = 0) = 10/40 = 1/4
P(Xj,v = 0)=40/80 = ½

$$M(Y|X_{j,v}) = \frac{1}{2}\cdot\frac{1}{4} + \frac{1}{2}\cdot\frac{1}{4} = \frac{1}{4}$$

Classifier 2

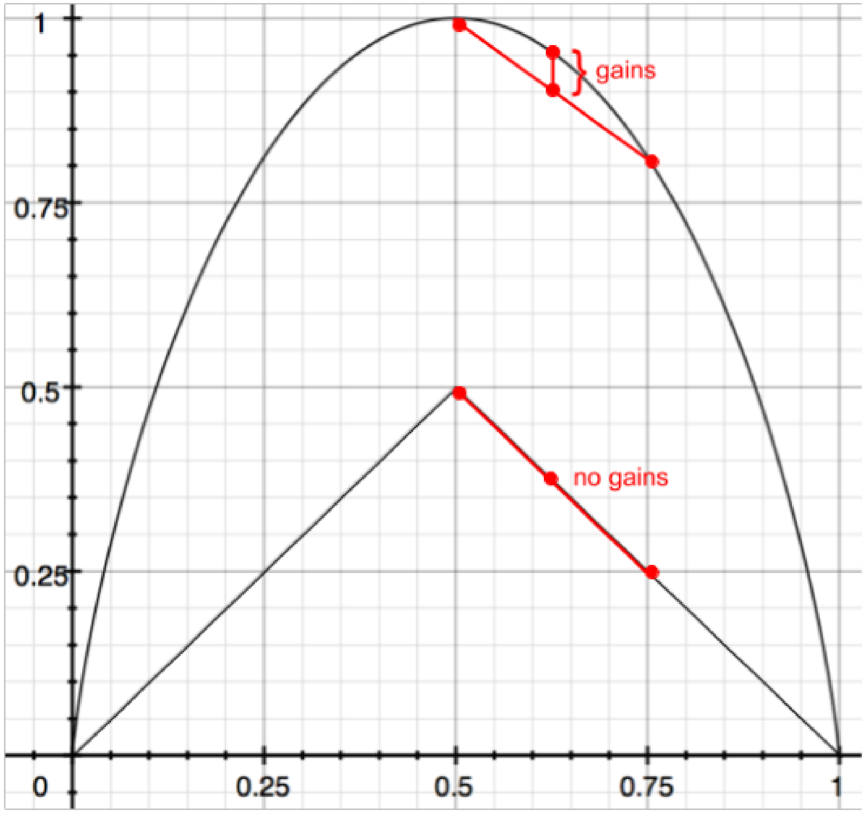
Set 1(A): 40 of class A and 20 of class B
M(Y_1|Xj,v = 1) = 20/60 = 1/3
P(Xj,v = 1)=60/80 = 3/4

Set 2(B): 0 of class A and 20 of class B
M(Y_2| Xj,v = 0) = 0/40 = 0
P(Xj,v = 0)=20/80 = ¼

$$M(Y|X_{j,v}) = \frac{3}{4}\cdot\frac{1}{3} + \frac{1}{4}\cdot 0 = \frac{1}{4}$$

Classification Losses (strict concavity)

This property means that the graph of the function lies strictly below the tangent line at every point (except the point of tangency, of course).


gains
no gains