CO395 Group 57: Dongxiao Huang, Zheng Xun Phang, Yufeng Zhang

Implementation

How do we select the best attribute at each node? For all attributes, we compute the information gain if the dataset is split on that attribute:

$$\text{Gain(Attribute)} = \mathcal{I}(p,n) - \left[\frac{p_0+n_0}{p+n}\mathcal{I}(p_0,n_0) + \frac{p_1+n_1}{p+n}\mathcal{I}(p_1,n_1)\right]$$

$$\begin{aligned} p_k &= \text{Number of positive examples with attribute} = k \\ n_k &= \text{Number of negative examples with attribute} = k \\ p = p_0 + p_1 &= \text{Number of positive examples before split} \\ n = n_0 + n_1 &= \text{Number of negative examples before split} \end{aligned}$$

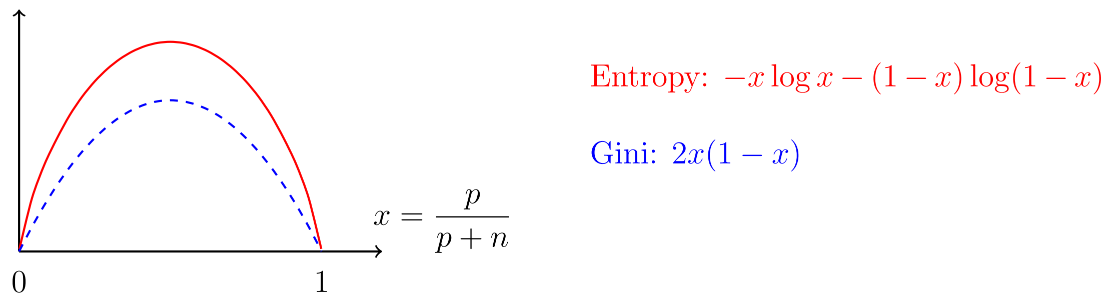
There are two common ways to measure information.

Entropy: $\mathcal{I}(p,n) = -\frac{p}{p+n}\log\frac{p}{p+n} - \frac{n}{p+n}\log\frac{n}{p+n}$ if $p,n \neq 0$

$\mathcal{I}(p,0) = \mathcal{I}(0,n) = 0$

Gini impurity: $\mathcal{I}(p,n) = \frac{p}{p+n}\left(1-\frac{p}{p+n}\right) + \frac{n}{p+n}\left(1-\frac{n}{p+n}\right)$

We used entropy since it's stated in the specification, but Gini impurity is faster to compute. Both metrics should give similar results since their graphs have a similar shape:



N.B. The *height* of those graphs is irrelevant because minimizing a function is equivalent to minimizing any positive multiple of that function. What matters is their *shape*.

The decision tree algorithm is given in the specification, so it's unnecessary to repeat it here. We try to use NumPy functions instead of Python loops for performance since the former run on vectorized C code.

To evaluate our decision tree, we performed K-fold cross validation as follows:

1. Shuffle the dataset and split it into $K = 10$ parts
2. For each $k \in \{1, \cdots, K\}$ we train the decision tree on the dataset *excluding* part k and then test the decision tree on part k. During testing, we aggregate the predictions from 6 trees (more details later) and increment the relevant cells in the confusion matrix.

Evaluation

Each cell of the confusion matrix is a total, not an average, over all folds of cross validation.

Predicted	Actual					
	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	106	11	3	5	5	1
Disgust	23	157	1	5	9	3
Fear	17	2	77	2	7	13
Happiness	22	6	1	179	6	1
Sadness	34	11	3	4	78	2
Surprise	27	3	7	4	4	161

We add up the relevant cells in the confusion matrix above to compute these summary statistics:

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Precision	0.809	0.793	0.653	0.833	0.591	0.782
Recall	0.463	0.826	0.837	0.899	0.716	0.890
F_1 score	0.589	0.809	0.733	0.865	0.647	0.832

$$\text{Classification rate} = \frac{106 + 157 + \cdots + 161}{1000 - (106 + \cdots + 161)} = 0.758$$

N.B. It makes no sense to compute classification rate for each emotion. For example, if the actual emotion is Anger then (Fear, Surprise) counts as a true negative.

The F score by van Rijsbergen (1975) is a harmonic mean of precision P and recall R, but there are two versions:

$$\frac{1}{F_\alpha} = \frac{\alpha}{P} + \frac{1-\alpha}{R} \qquad \text{for } 0 < \alpha < 1$$

$$\frac{1}{F_\beta} = \frac{1}{1+\beta^2}\frac{1}{P} + \frac{\beta^2}{1+\beta^2}\frac{1}{R} = (1+\beta^2)\frac{PR}{\beta^2 P + R} \qquad \text{for } \beta \in \Re$$

Our 6 trees classify the emotions Disgust, Happiness and Surprise with high accuracy, while Anger and Sadness seem harder to recognize.

Miscellaneous

Noisy-Clean Datasets Question

The noisy dataset has lower performance.

Ambiguity Question

In case our 6 trees predict that a given image depicts more than one emotion, we considered several methods of selecting only one emotion:

1. Select at random e.g. select the first emotion in alphabetical order

2. Disable each active unit in turn, and take a majority vote
Example:

Pruning Question