Chapter 1

Splitting Criterion

1.1 Gini index

The gini impurity is a decision tree splitting metric used by the CART[1] algorithm. In decision trees used for classification, the gini index is used to compute the impurity of a data partition. Given a training set S and a target attribute that takes on k different values (classes), the gini index $\mathcal{G}$ of set S is defined as,

$$\begin{aligned}\mathcal{G}(S) &= \sum_{i=1}^{k} p_i(1-p_i) \\ &= \sum_{i=1}^{k}(p_i - p_i^2) \\ &= \sum_{i=1}^{k} p_i - \sum_{i=1}^{k} p_i^2 \\ &= 1 - \sum_{i=1}^{k} p_i^2\end{aligned}$$

where p_i is the probability of an item chosen at random from the training set belonging to class i. If a subset has only 1 class, its gini index is 0 ($= 1 - 1^2$), such a set is a pure dataset. On the other hand if the class distribution is balanced i.e. probability of an item belonging to class i is $1/k$, its gini index achieves the maximum.

The gini splitting criterion requires the computation of a gini gain $\hat{\mathcal{G}}$ for each feature f.

Let feature f take on m unique values in $\mathbb{R}$. For each unique value $f_j, j = 1, ...m$ the gini gain $\hat{\mathcal{G}}(f_j, S)$ is computed as,

$$\begin{aligned}\hat{\mathcal{G}}(f_j, S) &= \mathcal{G}(S) - \mathcal{G}(f_j, S) \\ &= \mathcal{G}(S) - \left[\frac{|S_{left}|}{|S|}\mathcal{G}(S_{left}) + \frac{|S_{right}|}{|S|}\mathcal{G}(S_{right})\right]\end{aligned}$$

S_{left} and S_{right} are the partitions resulting from splitting the set on the basis of feature value f. S_{left} represents the set with feature value $f < f_j$ and S_{right} represents the set with feature value $f > f_j$. The feature f and value f_j that maximizes the gini gain $\hat{\mathcal{G}}$ are chosen as the splitting criterion at each internal node.

[1] Discussed in section **??**

1.2 Entropy

Entropy as a splitting metric is used by ID3, C4.5 and C5.0 tree algorithms. As the name suggests it is based on the concept of entropy in information theory. The entropy of a random variable is a measure of uncertainty and is mathematically defined by Shannon as,

$$H(X) = \sum_{i=1}^{n} P(x_i)I(x_i) = -\sum_{i=1}^{n} P(x_i)log_b P(x_i) \tag{1.1}$$

where X is a discrete random variable which takes values in $\{x_1, ..., x_n\}$, b is the base of the logarithm used, in Shannon entropy $b = 2$ to represent encoding using bits. $I(\bullet)$ is a measure of information content for x_i and is encoded in terms of the logarithm function.

The rationale behind using the logarithm function as a measure of information content is that it is additive for independent events. If event 1 occurs with probability p_1, $I(p1p2) = I(p1) + I(p2)$. If event 1 can have one of n equally likely outcomes and event 2 can have one of m equally likely outcomes then there are mn possible outcomes of the joint event with probability p_1p_2. $log_2(n)$ bits are needed to encode the first event and $log_2(m)$ bits are needed to encode the second event then $log_2 mn = log_2(m) + log_2(n)$ bits are needed to encode both. Any function that encodes information content should preserve this additivity, hence the choice is logarithmic .i.e. $I(p) = log(1/p)$.

Information gain under the entropy metric is defined as,

$$IG(T, f) = H(T) - H(T|f) \tag{1.2}$$

where T is a set of training samples, H is the entropy of the parent training set and $H(T|f)$ can be thought of as the weighted entropy of the left and right partition sets induced by a partition on the feature value of f. Let f take m unique values in $\mathbb{R}$. For each unique value $f_j, j = 1, ...m$ the information gain $IG(T, f_j)$ is computed as,

$$IG(T, f_j) = H(T) - \left[\frac{|T_{left}|}{|T|} H(T_{left}) + \frac{|T_{right}|}{|T|} H(T_{right}) \right] \tag{1.3}$$

where $H(T) = -\sum_{i=1}^{k} p_i log_2 p_i$ in the presence of k classes and p_i is the probability of a sample chosen at random belonging to class i.

Intuitively, both the gini gain and entropy splitting criteria can be thought of as metrics that measure the reduction in impurity from a split and select a split that maximizes this reduction.

1.2.1 Mathematical Formulation

Given input feature vectors $\{\mathbf{x}_i\} \in \mathbb{R}^d$ and a target variable $y_i \in \{0, 1\}$, a DT recursively partitions the training set at each node.

Without loss of generality, let the data at node q be represented by Q. The DT considers for each candidate split $\phi = (f, f_j)$ where f is a feature and f_j a threshold, partitions of the data Q into left and and right sets Q_l and Q_r such that,

$$Q_l(\phi) = \{\mathbf{x}_i \in Q : \mathbf{x}_i \leqslant f_j\}$$
$$Q_r(\phi) = \{\mathbf{x}_i \in Q : \mathbf{x}_i > f_j\}$$

The impurity denoted by $\mathcal{E}(\bullet)$ at node q is computed for all valid candidate splits ϕ on Q as,

$$\mathcal{S}(Q, \phi) = \frac{Q_l}{Q}\mathcal{E}(Q_l(\phi)) + \frac{Q_r}{Q}\mathcal{E}(Q_r(\phi)) \tag{1.4}$$

The candidate set ϕ that minimizes the sum of impurities of left and right sets is chosen as the parameter for the split.

$$\phi = \operatorname{argmin}_{\phi} \mathcal{S}(Q, \phi) \tag{1.5}$$

These steps are applied recursively for sets Q_l and Q_r to grow the tree until one of the stopping criteria are triggered or all the samples in the node belong to the same class.