1 Entropy

Entropy is a measurement of the uncertainty in a random variable.

$$H(Y) = -\sum_{y} P(Y = y)\log P(Y = y)$$

Example If a random variable Y can take two values 0 and 1 with the probability $P(Y = 0) = 0.7$ and $P(Y = 1) = 0.3$. The entropy of its distribution is

$$H(Y) = -0.3 \times \log 0.3 - 0.7 \times \log 0.7$$

The entropy of a distribution is 0 if the probability of Y taking any value is 1, i.e. $P(Y = y) = 1$.

$$H(Y) = -1.0 \times \log 1.0 - 0.0 \times \log 0.0$$

1.1 Conditional Entropy

is the expected value of the entropy of Y given X, over all values of X. This lets us quantify the entropy of Y given that we know X.

$$H(Y|X) = \sum_{x} P(X = x)H(Y|X = x)$$

Example Let X and Y be two random variables. We make the following observations for $\{X, Y\}$: {0,1}, {1,1}, {1,0}, {1,1}, {0,0}. What is the entropy of $H(Y)$? What is the conditional entropy of $H(Y|X)$?

1.2 Cross Entropy

The cross-entropy of the distribution Q relative to a distribution P over a given set is defined as follows:

$$H(P, Q) = -\sum_{x} P(X = x) \log Q(X = x)$$

Note that $H(P, Q) \neq H(Q, P)$.

Example Cross entropy is commonly used as the loss function in classification tasks. For example, in a classification task, a data has the label "True", i.e. $P(Y = \text{True}) = 1$. The predicted distribution of the labels is $\hat{P}(Y = \text{True}) = 0.7$ and $\hat{P}(Y = \text{False}) = 0.3$. What is the cross entropy loss $H(P, \hat{P})$ of the predicted distribution?

2 Mutual Information and Decision Tree

Mutual information is a measurement of the information we gain about Y by observing X. We get this by finding the difference between the entropy of Y, and the conditional entropy of Y given X.

$$I(Y;X) = H(Y) - H(Y|X)$$

What does it mean if the mutual information $I(Y;X)$ is large?

2.1 ID3

Algorithm: Split at the random variable X_i that maximizes the mutual information $H(Y;X_i)$.

Example You are trying to figure out whether your roommate will attend recitations using some attributes you know about them which may have affected their attendance in the past:

- The amount of sleep they had the night before the recitation: {Low,Medium,High}
- Whether they have a homework due in the next few days: {Yes, No}
- Did they understand the lectures during the week of the recitation: {Yes, No}
- Whether they attended the recitation: {Yes, No}

You have gathered the following data and decide that you want to make a decision tree to predict whether your roommate will attend today's recitation or not.

Sleep Amount	Homework Due Soon	Understand the Lectures	Attended Recitation?
Low	Yes	No	Yes
Medium	No	No	Yes
High	No	No	Yes
High	Yes	No	No
Low	Yes	Yes	No
Medium	No	Yes	No
Low	No	No	Yes
Low	No	No	No

1. What is the uncertainty of the label Attended Recitation (Y)?

2. What is the information gain of Sleep Amount (X_1)?

3. What is the information gain of Homework Due Soon (X_2) and Understand the Lectures (X_3)?

4. What does the ID3 algorithm put at the root of our decision tree?

3 Adaboost

AdaBoost relies on building an ensemble of weak learners, assigning them weights based on their errors during training.

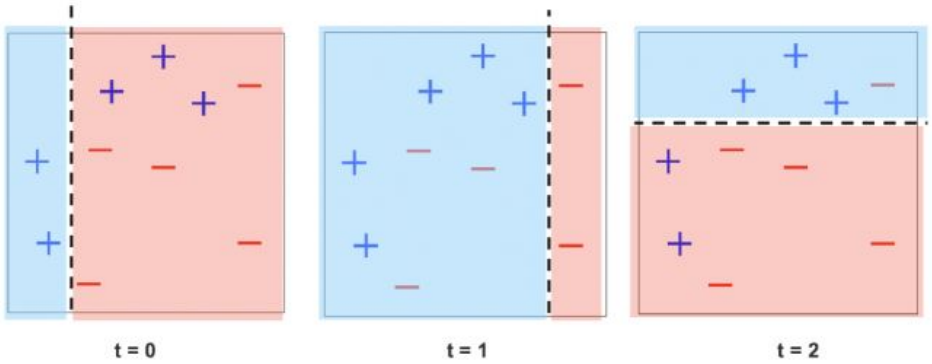
1. In the binary classification setting, what condition do we want on the error ϵ_t of weak learner h_t?

2. What happens to the weight $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ of classifier h_t if this condition is not met?

Given the above condition on all weak learners, the bound on training error decreases exponentially fast in the number of iterations T.

3.1 Example

The graphs below show three iterations running Adaboost with a depth 1 decision tree. Each dashed line represents the decision boundary of h_t, and the shaded regions represent the predictons, positive (blue) or negative (red). For each iteration find the weighted training error ϵ_t and importance α_t of h_t. For $t = 0$ and $t = 1$ also find the weight normalization Z_t and record the updated weight for each point.



(Adapted from Eric Xing's 10701 slides)

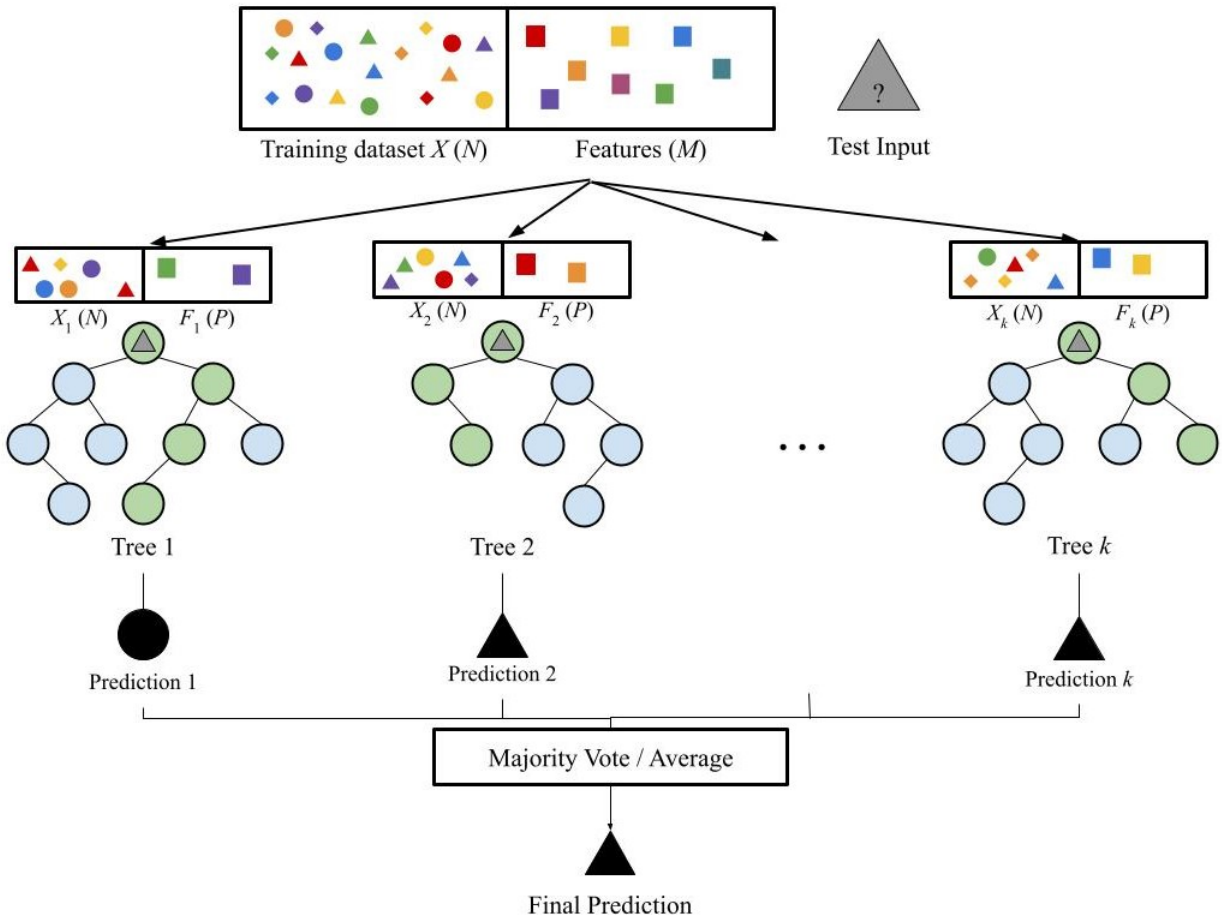
1. **t = 0**

2. $\mathbf{t = 1}$

3. $\mathbf{t = 2}$

4 Random Forest

Random forests have multiple individual decision trees that operate as an ensemble. Each individual tree in the random forest votes on a class prediction, and the class with the most votes becomes our model's prediction.



Ideally, these decisions trees will have low correlation so that they protect each other from their individual errors. To do this, we can use:

1. **Bagging:** Decisions trees are very sensitive to the data they are trained on, so small changes to the training set can result in significantly different trees. If each individual tree randomly samples from the dataset with replacement, it will result in different trees. Note that we are not splitting our dataset into different subsets and training each tree on a different subset. Instead, if we have a sample of size N, we are training each tree with a random sample of size N with replacement.

2. **Feature Randomness:** In a normal decision tree, when it is time to split a node, we use information gain and split on features that will give us the most separation in our data. However, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and results in lower correlation across trees and more diversification. Generally, if there are M total features, each decision tree will use $\sqrt{M}$ features.

Random forests can also be used for regression, like decision trees. For regression, we would get the average of all of the predicted values from the individual trees. Random forests, in practice, are favored over decision trees. Decisions trees tend to overfit, especially if they are very deep. Since a random forest uses multiple decision trees, it reduces the variance of the model and has better generalization. Random forests also still have almost all of the advantages of decision trees, though random forests are less interpretable and require more computation.