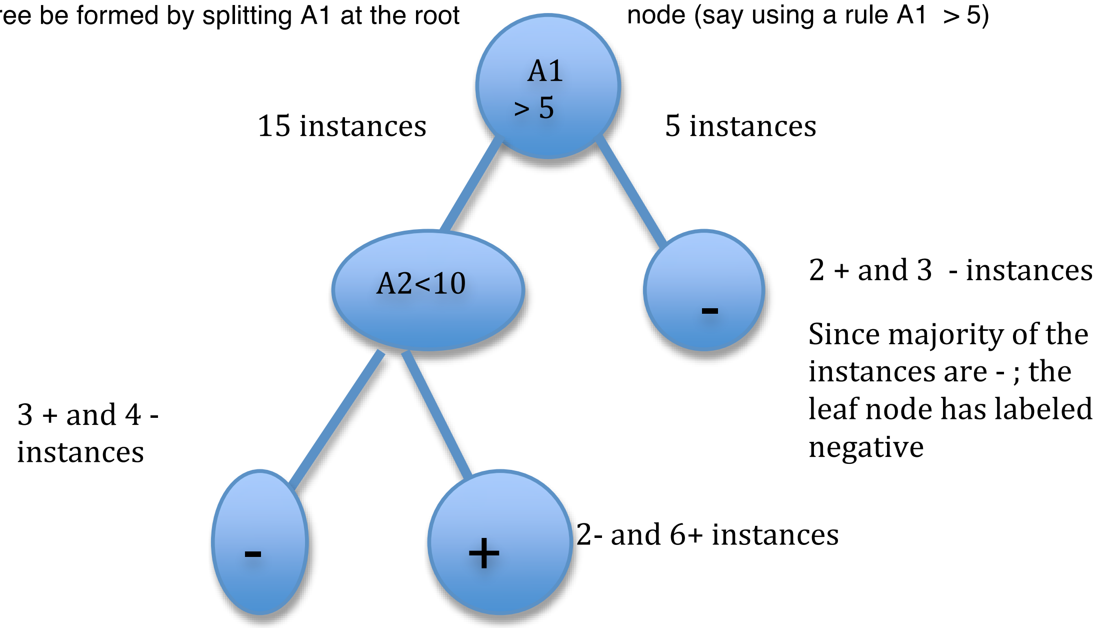
Assume that the dataset has 20 instances and 2 attributes A1 and A2. Let the following decision tree be formed by splitting A1 at the root node (say using a rule A1 > 5)



Estimating probabilities in our new algorithm

1. Consider the root node (A1>5). This splits the instances into two sets – one containing 15 and another containing 5 instances. If the path along 5 instances is followed, then probability of this 'rule' is 5 / 20 = 0.25. In addition, there are 2+ and 3- instances at the leaf node. Typically , to estimate probabilities of positive examples we would just do p^+ (at this node) = 2 / 5 and p- = 3/5. Instead, now we want to do the following: 5/ 20 * 2 / 5 for the positive examples and 5 / 20 * 3 / 5 for the negative examples. I worked out the above simple example and saw that the probabilities indeed sum to 1, so this is legal. Work out the above example to estimate probabilities at all the leaf nodes.
2. Instead of doing p+ = n+ / n; Now use Laplace and M-estimate formulas to work out the probability values using the scheme described above i.e. taking into account the probability of the 'rule'. Verify that the probabilities sum to 1 (or close to 1) in all the above cases. Provide a comparative chart of probability estimates obtained by different techniques.
3. In the above, note ties are not resolved. Use a Gaussian Kernel to do Kernel Density Estimation at the leaf nodes.
4. Implement the above technique in WEKA – (a) You should first modify the code to take into account the probability of the rules (b) You already have laplace and m-estimation implemented. Implement Kernel density estimation with a Gaussian Kernel for starters (c) Compare results on Iris or Spam-base dataset for the following techniques (1) Probability estimates obtained at leaf nodes using p+ = n+/n and p- = n-/n (2) Laplace correction (3) M-estimation (4) Taking Rule probabilities into account and modified P+ = n+ /n ; Laplace and M-estimation.