# Gradient Boosting Decision Trees: a minimal demo

Application: determine whether a loan is safe, a binary classification problem.

<https://www.coursera.org/learn/ml-classification/supplement/3TYwk/boosting-a-decision-stump>

## Choose a loss function

For binary classification, a commonly used robust loss function is binomial deviance



where represents two classes and the class label predication is .

## Compute the negative gradient of the loss function w.r.t



## Fit a decision stump to the negative gradient (i.e., replacing the response variable with the negative gradient)

Pseudo residual is the negative gradient



Fit to for the all training data. Note here the gradient is continuous instead of binary. Therefore, the decision stump should be a regression one. That is, to determine the splitting feature, we use the variance as the impurity metric. The intuition is that for each terminal node (leaf) the predicated value is just the mean of all samples assigned to that node. To minimize predication error in a squared-error loss function, the response variables in that terminal should be close to their mean value, i.e., a small variance.

## Update the Boosting ensemble



where is a shrinkage parameter, usually . For the initial value of , we can just assign it a constant like 0.

Theoretically, to mimic the steepest descent numerical algorithm, we should first find the weight or step size of the new component like . However, since we introduce the shrinkage for regularization purpose, there is actually no need to determine . The cost is that we may have to run more iterations since now our descent is not *steepest*. The final additive model is

