Choosing a Machine Learning Classifier

[[Link](http://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/)]

How large is your training set?

* Small training set:
  + High bias/low variance classifiers
    - Naïve Bayes
  + over low bias/high variance classifiers
    - kNN
  + since the latter will overfit
* low bias/high variance classifiers start to win out as your training set grows
  + they have lower asymptotic error
  + since high bias classifiers are not powerful enough to provide accurate models
* You can also think this as a generative model vs. discriminative model distinction

Advantages of some algorithms

* Naïve Bayes:
  + **Super simple**
  + Naïve Bayes **classifier will converge quicker** than discriminative models like logistic regression, so you need less training data.
    - provided the NB conditional independence assumption holds
  + Disadvantage:
    - it cannot learn interactions between features
    - Must be careful with correlated features
* Logistic Regression:
  + Lots of ways to regularize
  + Do not have to worry about features being correlated
  + Nice probabilistic interpretation, unlike decision trees and SVMs
  + Easy to update model to take in new data
    - using an online gradient descent method
* Decision Trees:
  + East to interpret and explain (for some people)
  + Easily handle feature interactions
  + Non-parametric
    - So, do not have to worry about outliers or whether the data is linearly separable
  + Disadvantage:
    - do not support online learning
    - must rebuild the tree when new examples come on
    - easily overfit
      * ensemble models like Random Forest (or boosted trees) help here
  + Random Forests are often the winner for lots of problems in classification (usually slightly ahead of SVMs)
  + Fast and scalable
  + Do not have to worry about touching a bunch of parameters (like you do with SVMs)
* SVMs:
  + High accuracy, nice theoretical guarantees regarding overfitting
  + Can work well even data is not linearly separable in the base feature space, given an appropriate kernel
  + Popular in text classification problems where very high-dimensional spaces are the norm
  + Disadvantage:
    - Memory-intensive
    - Hard to interpret
    - Cumbersome to run and tune