Gotchas!

Random forest almost always boosts your scoring accuracy above that of a single decision tree. Like other ensemble methods, they are effortlessly parallelizable, because you can have independent trees being trained on distinct cores and only need to get a mode vote of their classifications once you attempt to predict or score your data.

Two drawbacks of random forest are that since you're now planting an entire forest as opposed to a single tree, both training and prediction execution times suffer tremendously. Particularly, training is an order of magnitude more time consuming. You also lose the ability to inspect the resulting structure of your classifier as easily. No longer can you just print out a `.dot`-file flow chart since no single flowchart encompasses the forest, and your results aren't strictly based on following a single logic diagram. Along the same lines, the forest can no longer be linearized into IF...THEN blocks anymore.

Even with all that said, given the added accuracy performance-boost, if a high level of correctly classified samples is what you need, then moving from decision trees to random forest should be an easy choice to make.

Lecture: Random Forest
Quiz

Dive Deeper