As you say a lot has been discussed about this matter, and there's some quite heavy theory that has gone along with it that I have to admit I never fully understood. In my practical experience AdaBoost is quite robust to overfitting, and LPBoost (Linear Programming Boosting) even more so (because the objective function requires a sparse combination of weak learners, which is a form of capacity control). The main factors that influence it are:

* The "strength" of the "weak" learners: If you use very simple weak learners, such as decision stumps (1-level decision trees), then the algorithms are much less prone to overfitting. Whenever I've tried using more complicated weak learners (such as decision trees or even hyperplanes) I've found that overfitting occurs much more rapidly
* The noise level in the data: AdaBoost is particularly prone to overfitting on noisy datasets. In this setting the regularised forms (RegBoost, AdaBoostReg, LPBoost, QPBoost) are preferable
* The dimensionality of the data: We know that in general, we experience overfitting more in high dimensional spaces ("the curse of dimensionality"), and AdaBoost can also suffer in that respect, as it is simply a linear combination of classifiers which themselves suffer from the problem. Whether it is as prone as other classifiers is hard to determine.

Of course you can use heuristic methods such as validation sets or *k*

-fold cross-validation to set the stopping parameter (or other parameters in the different variants) as you would for any other classifier.