SNS COLLEGE OF TECHNOLOGY

Coimbatore-37.
An Autonomous Institution

COURSE NAME : 19CST301 & INTRODUCTION TO MACHINE LEARNING

III YEAR/ V SEMESTER

UNIT – 3 DEEP LEARNING

Topic: Regularization

Mrs.S.R.Janani
Assistant Professor
Department of Computer Science and Engineering

Regularization

- Regularization Besides the size of the constituent trees, J, the other meta-parameter of **gradient boosting is the number of boosting iterations M.**
- Each iteration usually **reduces the training risk** L(f_M), so that for M large enough this risk can be made arbitrarily small.
- However, **fitting the training data** too well **can lead to overfitting**, which degrades the risk on future predictions.
- Thus, there is an **optimal number M∗ minimizing future risk** that is application dependent.
- A convenient way to estimate M∗ is to **monitor prediction risk** as a function of M on a validation sample.
- The value of M that **minimizes this risk** is taken to be an estimate of M∗ .
- This is analogous to the **early stopping strategy** often used with neural networks

Regularization

- **Shrinkage**
- **Subsampling**

Shrinkage

- Controlling the value of M is not the only possible regularization strategy.
- As with ridge regression and neural networks, shrinkage techniques can be employed as well (see Sections 3.4.1 and 11.5).
- The simplest implementation of shrinkage in the context of boosting is to scale the contribution of each tree by a factor 0 < ν < 1 when it is added to the current approximation.
- That is, line 2(d) of Algorithm 10.3 is replaced by

$$f_m(x) = f_{m-1}(x) + \nu \cdot \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm}).$$

Shrinkage – cont.

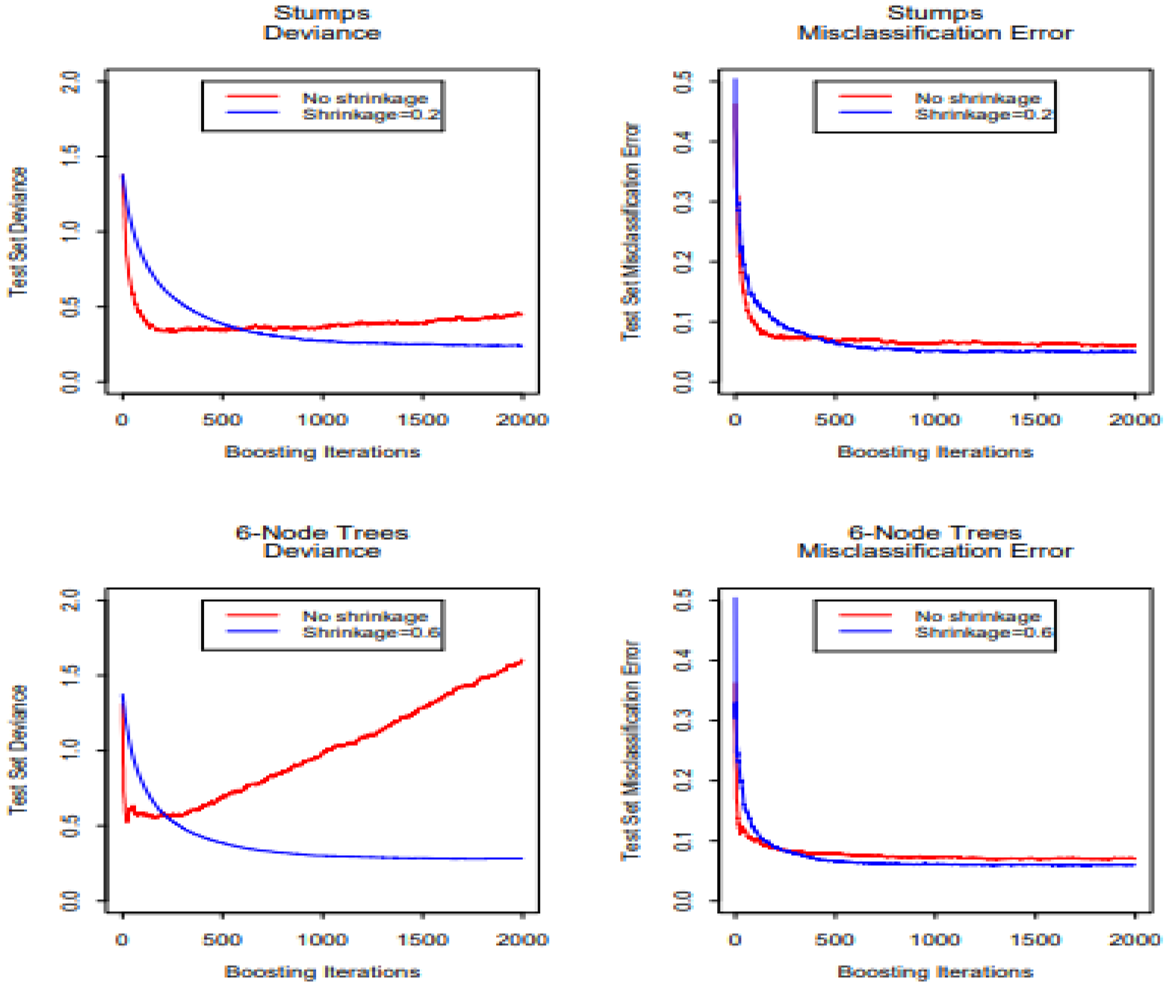
- The parameter "**v**" can be regarded as **controlling the learning rate** of the boosting procedure. Smaller values of v (more shrinkage) **result in larger training risk** for the same number of iterations M.
- Thus, both **v and M control prediction risk on the training data**.
- However, these parameters do not operate **independently.**
- Smaller values of v lead to larger values of M for the same training risk, so that there is a tradeoff between them.
- Empirically it has been found (Friedman, 2001) that smaller values of v favor better test error, and require correspondingly larger values of M.

Shrinkage – cont.

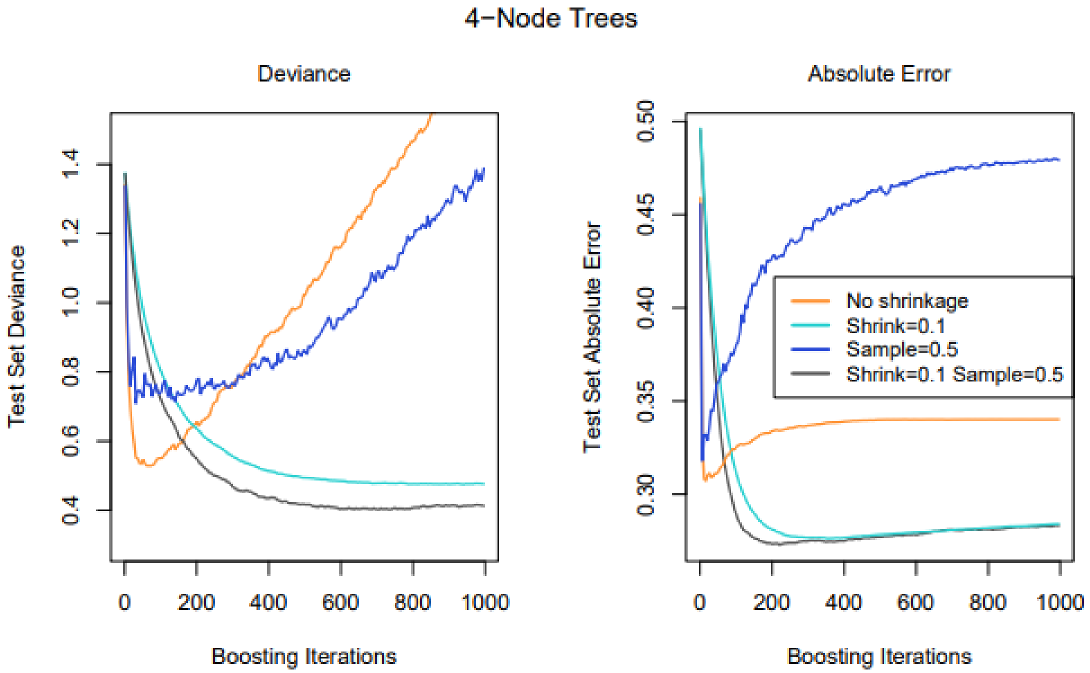
- In fact, the **best strategy appears to be to set ν to be very small** (ν < 0.1) and then choose M by early stopping.
- This yields dramatic improvements (over no shrinkage ν = 1) for regression and for probability estimation.
- The corresponding improvements in misclassification risk via (10.20) are less, but still substantial. The price paid for these improvements is computational:
  - smaller values of ν give rise to larger values of M, and
  - computation is proportional to the latter.
- However, as seen below, many iterations are generally computationally feasible even on very large data sets.
- This is partly due to the fact that small trees are induced at each step with no pruning. Figure 10.11 shows test error curves for the simulated example (10.2) of Figure 10.2.
- A **gradient boosted model (MART) was trained using binomial deviance**, using either stumps or six terminal-node trees, and with or without shrinkage.
- The benefits of shrinkage are evident, especially when the binomial deviance is tracked.
- With shrinkage, each test error curve reaches a lower value, and stays there for many iterations. Section 16.2.1 draws a connection between forward stagewise shrinkage in boosting and the use of an L1 penalty for regularizing model parameters (the “lasso”). We argue that L1 penalties may be superior to the L2 penalties used by methods such as the support vector machine.

Subsampling

- The bootstrap averaging (bagging) improves the performance of a noisy classifier through averaging.
- We can exploit the same device in gradient boosting, both to **improve performance and computational efficiency.**
- With **stochastic gradient boosting** (Friedman, 1999), at each iteration we sample a fraction η of the training observations (without replacement), and grow the next tree using that subsample.
- The rest of the algorithm is identical. A typical value for η can be 1/2 , although for large N, η can be substantially smaller than 1/2.
- Not only does the sampling reduce the computing time by the same fraction η, but in many cases it actually produces a more accurate model.
- Figure 10.12 illustrates the effect of subsampling using the simulated example (10.2), both as a classification and as a regression example.
- We see in both cases that sampling along with shrinkage slightly outperformed the rest. It appears here that subsampling without shrinkage does poorly.



- FIGURE 10.11. Test error curves for simulated example (10.2) of Figure 10.9, using gradient boosting (MART). The models were trained using binomial deviance, either stumps or six terminal-node trees, and with or without shrinkage. The left panels report test deviance, while the right panels show misclassification error. The beneficial effect of shrinkage can be seen in all cases, especially for deviance in the left panels.



- FIGURE 10.12. Test-error curves for the simulated example (10.2), showing the effect of stochasticity. For the curves labeled "Sample= 0.5", a different 50% subsample of the training data was used each time a tree was grown. In the left panel the models were fit by gbm using a binomial deviance loss function; in the right-hand panel using square-error loss.

The downside is that we now have four parameters to set: J, M, ν and η. Typically some early explorations determine suitable values for J, ν and η, leaving M as the primary parameter.

References

- AlpaydinEthem, “Introduction to Machine Learning”, MIT Press, Second Edition, 2010.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, Springer; Second Edition, 2009.

Thank you