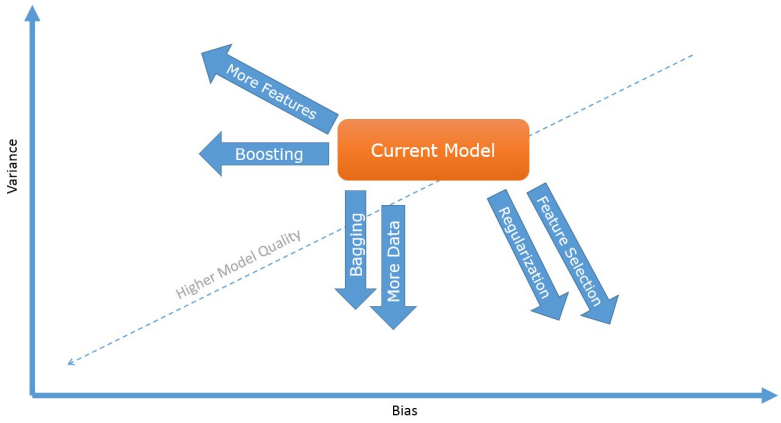
Ahmed El Deeb Follow
Relevance engineer. Machine Learning practitioner and hobbyist. Former entrepreneur.
May 28, 2015 · 4 min read

7 Ways to Improve your Predictive Models



This is a figure I dug up from an old slide deck I prepared years ago for a workshop on predictive modeling. it illustrates what I think of as the "war horse" of model tuning ('cause you know, it kind of looks like a horse, with an extra spear). It also is a kind of map for navigating the Bias-Variance space.

Top highlight

Bias and **variance** are the two components of imprecision in predictive models, and in general there is a trade-off between them, so normally reducing one tends to increase the other. **Bias** in predictive models is a measure of **model rigidity** and**inflexibility**, and means that your model is not capturing all the signal it could from the data. Bias is also known as **under-fitting**. **Variance** on the other hand is a measure of **model inconsistency**, high variance models tend to perform very well on some data points and really bad on others. This is also known as **over-fitting**and means that your model is **too flexible** for the amount of training data you have and ends up picking up noise in addition to the signal, learning random patterns that happen by chance and do not generalize beyond your training data.

The simplest way to determine if your model is suffering more from bias or from variance is the following rule of thumb:

If your model is performing really well on the training set, but much poorer on the hold-out set, then it's suffering from high variance. On the other hand if your model is performing poorly on both training and test data sets, it is suffering from high bias.

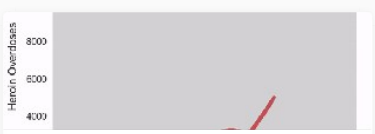
Depending on the performance of your current model and whether it is suffering more from high bias or high variance, you can resort to one or more of these seven techniques to bring your model where you want it to be:

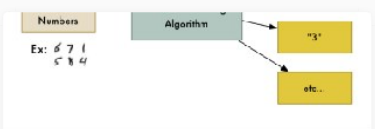
1. **Add More Data!** Of course! This is almost always a good idea if you can afford it. It drives variance down (without a trade-off in bias) and allows you to use more flexible models.
2. **Add More Features!** This is almost always a good idea too. Again, if you can afford it. Adding new features increases model flexibility and **decreases bias**(on the expense of variance). The only time when it's not a good idea to add new features is when your data set is small in terms of data points and you can't invest in #1 above.

3. **Do Feature Selection.** Well, … only do it if you have a lot of features and not enough data points. Feature selection is almost the inverse of #2 above, and pulls your model in the opposite direction (decreasing variance on the expense of some bias) but the trade-off can be good if you do the feature selection methodically and only remove noisy and in-informative features. If you have enough data, most models can automatically handle noisy and uninformative features and you don't need to do explicit feature selection. In this day and age of "Big Data" the need for explicit feature selection rarely arises. It is also worth noting that proper feature selection is non-trivial and computationally intensive.
4. **Use Regularization.** This is the neater version of #3 and amounts to implicit feature selection. The specifics are beyond the scope for this post, but regularization tells your algorithm to try to use as few features as possible, or to not trust any single feature too much. Regularization relies on smart implementations of training algorithms and is usually the much preferred version of feature selection.
5. Bagging is short for Bootstrap Aggregation. It uses several versions of the same model trained on slightly different samples of the training data to reduce variance without any noticeable effect on bias. Bagging could be computationally intensive esp. in terms of memory.
6. Boosting is a slightly more complicated concept and relies on training several models successively each trying to learn from the errors of the models preceding it. Boosting decreases bias and hardly affects variance (unless you are very sloppy). Again the price is computation time and memory size.
7. **Use a more different class of models!** Of course you don't have to do all the above if there is another type of models that is more suitable to your data set out-of-the-box. Changing the model class (e.g. from linear model to neural network) moves you to a different point in the space above. Some algorithms are just better suited to some data sets than others. Identifying the right type of models could be really tricky though!

It should be noted though that model accuracy (being as far to the bottom left as possible) is not the only objective. Some highly accurate models could be very hard to deploy in production environments and are usually black boxes that are very hard to interpret or debug, so many production systems opt for simpler, less accurate model that are less resource-intensive, easier to deploy and debug.

Machine Learning | Predictive Modeling | Data Science



Also tagged Data Science

Five books every data scientist should read that are not about...

Isaac Faber
5 min read

4.1K

Also tagged Data Science

How to Create Animated Graphs in Python

Viviane
5 min read

1.1K

Also tagged Machine Learning

Machine Learning is Fun!

Adam Geitgey
15 min read

50K

Responses

Write a response...

Show all responses

Never miss a story from **Rants on Machine Learning**, when you sign up for Medium. Learn more

GET UPDATES