Baselines

Introduction

Assume you work for an apparel retailer as a data scientist and your task is to send out 100,000 advertising mail pieces to past customers about a new line of fall apparel. You have access to a database of information about past customers, including what they have purchased and their demographic information (Ramakrishnan 2018). What model should you build first?

A *baseline*!

In this article, we will discuss what a baseline is and where it fits in our data analysis projects. We will see that there are two different types of baselines, one which refers to a simple model, and another which refers to the best model from previous works. A baseline guides our selection of more complex models and provides insights into the task at hand. Nonetheless, such a useful tool is not easy to handle. A literature review (Lin 2019; Mignan 2019; Rendle et al. 2019; Yang et al. 2019) shows that many researchers tend to compare their novel models against weak baselines which poses a problem in the current research sphere as it leads to optimistic, but false results. A discussion on such contemporary issues and open-ended questions is also provided at the end, followed by suggestions aimed at the community.

Baselines in Problem Solving – Overview

People use data analysis techniques to solve practical problems that may or may not require deep understanding of machine learning algorithms. Common-sense models that act as baselines in many cases perform surprisingly well compared to complicated machine learning models.

What is a baseline?

A baseline is a simple model that provides reasonable results on a task and does not require much expertise and time to build. Common baseline models include linear regression when predicting continuous values, logistic regression when classifying structured data, pretrained convolutional neural networks for vision related tasks, and recurrent neural networks and gradient boosted trees for sequence modeling (Ameisen 2018).

Because it is simple, a baseline is not perfect. For example, the bag of words model for language modeling does not take into account the order of the words occurring in each sentence, hence, it is missing out on a lot of structural information. Also, many baselines largely depend on heuristics embedded in the model or on strong modelling assumptions (i.e. linear models). Finally, from a researcher's perspective, a baseline does not provide cutting-edge research experience which makes it harder to publish.

If baselines are not perfect, then what makes them useful? Let's circle back to the apparel example. Common sense tells us to mail the top 100k loyal customers to maximize our expected return. Loyalty can be measured in terms of three features. If customers shop a lot, spend a lot, and have recently shopped with your store, then they are likely to be loyal (Ramakrishnan 2018). One way to combine these three pieces of information is binning, where we categorize each customer for each category and then simply rank them based on some scale. The top 100k customers are the ones you should mail. In fact, this model is a practical heuristic, called the Recency-Frequency-Monetary Heuristic, which is commonly used in database marketing and direct marketing. It is easy to create, easy to explain, easy to use, and effective.

Why start with a baseline?

A baseline takes only 10% of the time to develop, but will get us 90% of the way to achieve reasonably good results. Baselines help us put a more complex model into context in terms of accuracy. Ameisen (2018) names four levels of performance we can categorize a model into. Using these levels of performance we can determine what level we want our model to achieve based on our domain knowledge, our computational and human resources, and our business goals. Each level of performance will guide our selection of a baseline.

Trivially Attainable Performance

The Trivially Attainable Performance is the performance that can be obtained easily and you should expect your model to be better than this. For instance, in classification, guessing the most frequent class provides a quick solution.

Human Level Performance

> We use machine learning methods mainly due to two reasons: automate tasks, or perform tasks where machines and algorithms are faster and better than humans (AlphaGo). For the former, a model achieving Human Level Performance means that the model is ready for deployment and for the latter it is only a lower bound on the expected performance.
>
> **Reasonable Automated Performance**
>
> Reasonable Automated Performance represents what can be obtained with a relatively simple method. This benchmark is critical in assessing whether a complex model is performing well and in addressing the accuracy and complexity trade-off.
>
> **Required Deployment Performance**
>
> Finally, Required Deployment Performance is the minimal performance that makes your model ready for production from a business standpoint.
>
> (Ameisen 2018)

Another advantage of a baseline is that it's easy to deploy, it is faster to train as few parameters can quickly fit to your data, and it is often simple enough to allow easy problem detection. Whenever an error pops up, it is likely due to a bug, some defect of the dataset, or due to wrong assumptions. A baseline is quick to put into production as it does not require much infrastructure engineering, and it is also quick for inference because it has low latency.

What to do after building a baseline?

Having constructed a baseline, the next step is to see what the baseline fails to capture. This will guide our choice of a more complex model. That being said, we should also not neglect the fact that improving upon baselines can make already successful cases fail, as improvements are not strictly additive. This is especially true in the case of deep learning models where the lack of interpretability makes it harder to infer failure cases.

Furthermore, baselines help us understand our data better. For example, for a classification task we can get an idea of what classes are hard to separate from each other by looking at the confusion matrix. If we have a lasso regression problem, we can look at the non-zero coefficients to guide feature selection. Furthermore, if the model neglects some features that our domain knowledge tells us are important, then that sends a warning signal as our data may not be a good representation of the population or it could just be that the model is not suitable.

Baselines in Research – Weak Baselines

Now that we have an idea of what baselines are, let us shift our attention to how they have been used in academia. Great expectations surround AI and Machine Learning about their role in changing the world. The past few years have witnessed an increase in research output and empirical advancement in all Machine Learning related areas. However, this increase has not necessarily been matched with a rigorous analysis of the underlying methods. This has led to a gap between the level of empirical advancement and the level of empirical rigor which is partly due to the research culture that our community has developed; a culture that rewards "wins" which usually means beating a previous method on some task (Lin 2019). While there seems to be nothing wrong with this culture, issues emerge when it is the authors themselves who determine what previous method to compare against. For many, a "win" means beating a baseline, even if that baseline is weak. The following examples support this claim.

Netflix vs Movielens

Before we discuss "wins" we need to have a clear understanding of what it means for a baseline to be good. Rendle et al. (2019) show that measuring the performance of a baseline model for a particular task is not as straightforward as it might first seem. In their work, they refer to two extensively studied datasets, Movielens and Netflix, to indicate that running baselines properly is difficult.

The Movielens datasets are widely used in recommender systems research. Over the past five years, numerous new recommendation algorithms have been published, reporting large improvements over baseline models such as Bayesian MF, RSVD, and BPMF, as reported in Figure 2. Rendle et al. (2019) show that these baselines can beat the current state-of-the-art models on this benchmark just by tuning the baselines and combining them with simple, well-known methods. Furthermore, Bayesian MF, which was previously reported to perform poorly, after proper tuning is able to outperform any results reported on this benchmark so far.

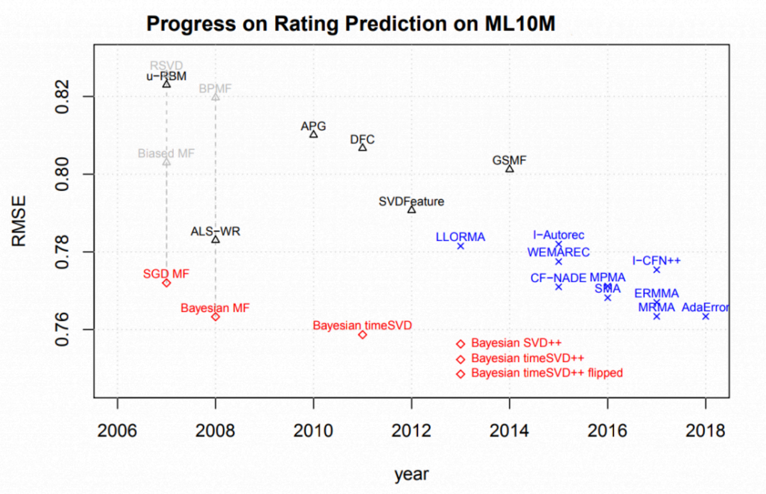

Figure 2. Rerunning baselines for Bayesian MF and other popular methods, where results marked in blue crosses were reported by the corresponding researchers and those marked as black triangles were run as baselines. Lower is better. After additional tuning, these baselines (marked in red) can achieve much better results than previously reported methods. Table borrowed from Rendle et al. (2019).

The Netflix Challenge also demonstrates that running methods properly is hard. The Netflix Prize was awarded to the team who improved Netflix's own recommender system by 10%. From the beginning of this competition, matrix factorization algorithms were widely used as baselines. Authors show some key results for vanilla matrix factorization models reported by top competitors and winners, and there is steady progress on the different methods based on the same baseline. This indicates that achieving good results even for a presumably simple method like matrix factorization is a non-trivial task and takes time and effort.

These findings on the differences between Movielens and Netflix datasets question the current results of the baseline models as well as any newly proposed methods. Even though these results follow good practices (conducting a reasonable hyperparameter search, reporting statistical significance and allowing reproducibility), they are still suboptimal (Rendle et al. 2019).

Weak Baselines in Information Retrieval

A similar example is presented by Lin, a professor at the University of Waterloo, who decided to test the effectiveness of some well-known, albeit old, ranking algorithms against two recently published papers that appear at top Information Retrieval conferences (2019). The old ranking algorithms have served as baselines for years, however, there has not been much interest in tuning them and discovering their full potential. What Lin discovers is that by exploring their hyperparameters via a simple grid search, one can find much better-performing baselines which nullify the improvements of the other novel methods.

Lin considers variants of BM25, Query Likelihood with Dirichlet Smoothing (QL) and RM3 as possible baselines and compares them against the reported performances in Paper 1 and Paper 2. Figure 3 lists the results.

Both Paper 1 and Paper 2 suffer from similar problems – weak baselines. Lin reports that the authors of Paper 1 do a good job tuning the baseline they use (QL). What they fail to do is consider other baselines which may potentially result in better performances. In this particular case, BM25+RM3 matches the performance of Model 1, rendering its claimed improvements invalid. Paper 2, on the other hand, uses an untuned version of BM25 and the tuned baseline surpasses Model 2 by a fair margin, which raises the question whether they are really making progress (Lin 2019).

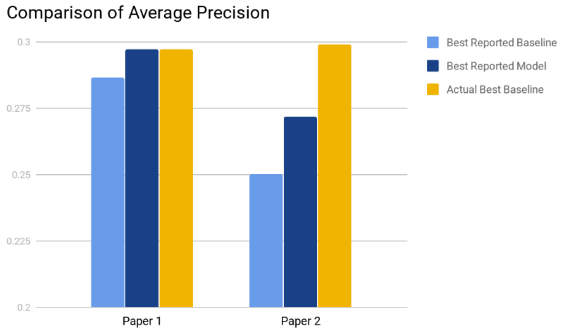

Figure 3. The chart compares the reported average precision (AP) of the baselines and the contributions of both Paper 1 and Paper 2 versus the AP of the tuned baselines. Results from Lin (2019).

Yang et al. take this issue to the next level and ask if such comparisons are common practice among other researchers (2019). They survey a multitude of papers in Information Retrieval to shed some light into this question. The results are deeply troubling. Figure 4 plots the average precision for every baseline used in the surveyed papers alongside that paper’s contributed performance. It is worth noting that, not only the baselines, but also 60% of the proposed models perform worse than an untuned RM3 (dashed black line). Furthermore, only six papers “win” over the best performing model of 2004 (solid black line), and out of those, five papers are more than a decade old. As Lin puts it, the more alarming fact is that this trend has gone largely unnoticed by the community (2019).

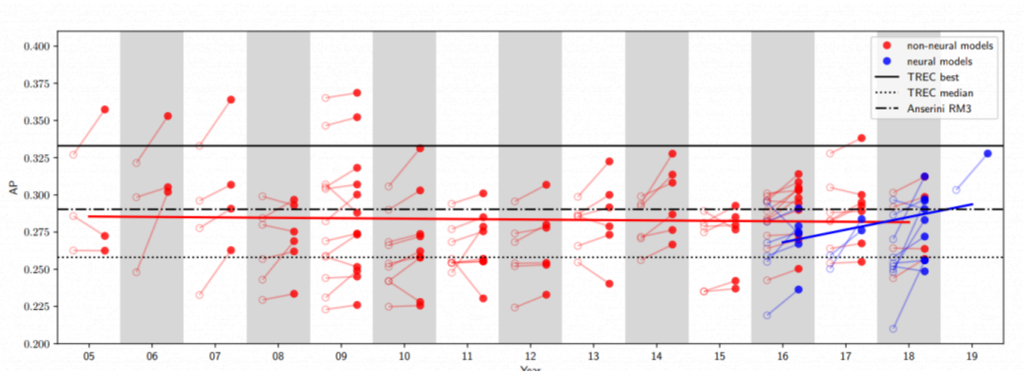

Figure 4. Average Precision of the baselines used in the papers surveyed by Yang et al. versus the year they were published. Empty circles represent the baseline and the linked red circles represent the contributed performance. Table borrowed from Yang et al (2019).

Nonetheless, this does not mean that the past decade has not witnessed any progress. It is sometimes acceptable to achieve similar or worse performances if it means trading that bit of accuracy for more robustness or interpretability. While the survey done above considers only the rates of average precision reported, it does not survey the novelty scores and contributions that these papers may have. Therefore, these results are somewhat overly pessimistic and should be taken with a grain of salt.

Construction and Improvement of Baselines

A recent paper, published in Nature by DeVries et al (2018), proposed a deep neural network (DNN) with 13k parameters to forecast aftershock locations in the aftermath of large seismic events. Interestingly, this DNN is outperformed by a much simpler baseline model. Mignan and Broccardo replicate these results by using a two-parameter logistic regression (2019). This demonstrates that the proposed deep learning strategy does not provide any new insights.

How can we avoid this mistake?

Improving Baselines Strategically

These ideas were originally presented in Merity (2017).

Do not add unnecessary complexity

A good rule of thumb is to start with a simple baseline, for example, logistic regression. The baseline should be stable in the sense that its results should be warranted up to some degree. Therefore, following Occam's Razor principle is most valid for a baseline. If adding a lot of complexity means a tiny increase in performance, then keep that complexity out of the baseline.

Find the limits of your baseline

After we have constructed a simple and fast baseline we get it into shape. This can be done by: extensively tuning parameters, adopting regularization techniques, conducting data processing etc. We can afford many runs tuning the hyperparameters because the baseline is simple. Similarly, bugs and flawed assumptions become easier to find as the model is not complex enough to hide them from us.

Gradually progress towards complex models

While we progress towards more complex models, it is important to understand if the complexity we add is warranted. A strategy that people follow is to keep increasing the model's parameters until it overfits on the dataset and then introduce regularization to prevent overfitting. Even though some work (Belkin 2019) has shown that further increasing the complexity of your model may lead to good generalization, this still leads to a lack of understanding of why increasing complexity works. Besides, simply adding complexity leads to too many moving parts, and it may be hard to isolate the improvements. A key idea when moving towards more complex models is to make sure we know what we are trading off. Add a component, one at a time, evaluate it, and then keep it or remove it. For example, we may have to weigh between component A which makes the model far slower but achieves gains, or component B that makes the model less general but faster.

Discussion

Now that we've discussed how to get the baseline right and strong, one question we should ask ourselves is how much complexity should we add to a baseline?

Baseline Complexity Revisited

Suppose we are doing sequence modeling in a research setting. Our baseline methods include a traditional recurrent neural network (RNN) and our novel model is based on the state-of-the-art transformer model. When we compare these models for a generation task, we consider the option of adding beam search to our modified transformer model. Should we also add beam search to the baseline RNN model? If we do not, is the comparison fair? On the other hand, if we do, is RNN still a baseline?

This question leads to two different notions of baselines. One is the baseline we use in problem solving, where we define it to be a very simple method that is known to perform reasonably well at some task. The second kind is the baseline models that we use in research, such as the current state-of-the-art models. This serves the purpose of showing that our novel models are performing better. It's important to be able to distinguish between these two use cases of baselines.

We have seen through the Movielens and IR examples how using an untuned baseline leads to issues in model comparison. In case we decide to use the second notion of a baseline, that of the current state-of-the-art, a follow up question could be "should we tune the state-of-the-art?" The answer depends. If we enhance our proposed models with general techniques such as regularization techniques like dropout or weight decay, then we should also implement them for the baseline for a fair comparison. Most importantly, although the ML community tends to criticize incremental improvements of already existing models, empirical results are still needed to explore different combinations of techniques to analyze and better understand their interactions.

In a Reddit thread (radarsat1 2019) that discusses the 2 parameter logistic regression versus a 13k deep learning model on aftershock prediction, we found a comment that made a good point. In essence, the commenter drew a comparison between statistics and deep learning. More specifically, he argues that both statisticians and deep learning engineers make claims based on experimental results, but statisticians have a rigorous process to show something is statistically significant with a certain confidence interval. The same question can be asked regarding baselines, namely how can we rigorously show that our complex models are doing better than the baselines? A look at the accuracy bar does not suffice to answer this question. When designing a new model, we should be constantly aware that we are not building a complex model for the sake of its complexity but for its expressiveness. Our inability to address this issue on both the intuitive and theoretical level is an intellectual debt that we will eventually need to pay off.

A Community Approach

The examples we have listed so far do not only suggest a problem with a specific group of self-centered researchers, but a problem with the entire community and the way the community assesses productivity. If papers that report comparisons against weak baselines keep getting accepted into top conferences, then perhaps the community is as guilty as the authors. Therefore, fundamental changes are needed in order to build a more serious and flourishing community of researchers. We suggest the following ideas borrowed from Lin (2019).

Procedural Changes in Conferences

Similar to the way we have developed policies against plagiarism we need to develop policies against weak baselines. These can take various forms, such as: asking reviewers to carefully examine the validity of the baselines used in submitted papers; asking the authors to confirm the quality of their baselines, by pointing out the way it was tuned or the source from which it was taken; or comparing against a popular well-tuned baseline if the area of research is not new.

Execution-Centric Leaderboard

Since authors keep rebuilding the baselines they use, a central leaderboard which keeps track of submitted models and compares them against a central baseline which has been well-tuned by the community might prove helpful. This way authors need not worry about tuning their baselines or finding the appropriate one to use.

Cultural Change

In the current community, paper count represents the top metric which measures the productivity of a researcher. What choice do researchers have when under such pressure, the only option is to prepare a solution that beats a weak baseline and publish that result? The old “quality over quantity” is most appropriate in this case. Therefore, unless we address this issue, there cannot be positive progress.

Nonetheless, no solution is perfect and there are issues that arise with each recommendation. For example, it is not clear how one determines what a strong baseline is if the research area is exotic and the number of researchers is limited. A leaderboard would fail in this case, and publishing committees would face a much harder task at validating the baselines used. Similarly, datasets such as those in medicine are usually private, making it infeasible to construct a good public baseline. In that case, we have no choice but to trust the owners of the dataset in their rigor. Finally, using a centric leaderboard may lead to issues of overfitting to the dataset, as all researchers will be comparing against the same baseline and that is also undesirable.

Conclusion

We have seen that a baseline is an important component of a data analysis project. A baseline helps us understand the task better, helps us discover inconsistencies with the data, and gets us up and running in no time. However, the use of a weak baseline may lead to the false belief that some methods are performing well when they are not, which hinders development and progress. Hence, we need to be cautious about the baseline we use and the amount of effort we pay to tuning. Finally, we should be aware that such weak comparisons are not solely the researchers' fault, but they suggest a defect in the way we do research as a community, which calls for changes from above. This will ultimately lead to a healthier research environment which appreciates good knowledge.