Visualizing the bias-variance tradeoff

Gursimran Singh
Graduate Student
Department of Computer Science
University of British Columbia
Email: msimar@cs.ubc.ca

Kate Melnykova
Graduate Student
Department of Mathematics
University of British Columbia
Email: melnykova@math.ubc.ca

Halldor Thorhallsson
Graduate Student
Department of Computer Science
University of British Columbia
Email: halldorb@cs.ubc.ca

1 Introduction

Machine learning is at the forefront of many technological breakthroughs of the last decade like speech recognition [1, 2], computer vision [3, 4], language understanding [5, 6] and driverless cars [7]. Since these breakthroughs happened in a relatively short span of time, existing pedagogical methods were inadequate to generate enough skill power, creating a demand-supply gap [8]. In the recent years, we have seen interesting attempts through MOOCs [9], blog articles [10] and intuitive visualizations [11, 12] to cover this gap. Despite these resources, the students often lack an intuitive understanding of the nuances of various machine learning concepts and algorithms, which behave quite differently with changing parameters.

For instance, nearly all machine learning algorithms allow a user to control the complexity of the model through parameters. The practitioners want their model to be complex enough to capture sophisticated relationships in the data while keeping it simple to prevent noise from affecting the outcome. This leads to a fundamental tradeoff known as the bias-variance tradeoff which is of paramount importance for optimal choice of the hyperparameters of the learning algorithms. Numerous machine learning resources have attempted to explain this fundamental concept which, albeit well presented, often lack the ability for a student to gain an intuitive understanding.

We argue that this lack of understanding is primarily due to the inability to play around with the parameters of algorithms. In order to give an intuitive understanding of the bias-variance tradeoff, we propose to develop interactive visualisations for a few classical machine learning algorithms. It will allow students to tinker and experiment with algorithms and their parameters, and in the process students will develop a strong understanding of the various machine learning concepts like the bias-variance tradeoff.

2 Personal Expertise

2.1 Gursimran Singh

Gursimran is interested in scaling computer science pedagogy to make it accessible to millions. During his undergraduate, he did an internship at IITB (IIT Bombay) where he developed interactive visualizations of numerical algorithms and statistics. Post that, he joined a startup as a research engineer to develop a scalable machine-learning based approach to grade computer programming assignments. Since he joined UBC as a computer science student he has been working as a TA for master of data science (MDS) program.

2.2 Kate Melnykova

Kate has been investigating the underlying theory of machine learning and big data analysis during her PhD studies. She is very knowledgeable about the subject matter, but she is very fresh to JavaScript and D3.js in particular. In addition, Kate has been an instructor for two math courses and a TA for numerous math courses, so she has gained lots of experience of explaining concepts to students.

2.3 Halldor Thorhallsson

Halldor has been developing statistical models and analyzing data during the course of his masters degree by taking courses, research projects and internships related to machine learning and is very familiar with this topic. He has also been a TA for numerous courses in Computer Science and Statistics.

3 Data And Tasks

This project aims to provide the visual, intuitive, and interactive insight into the relation between bias and variance for three ML techniques: linear regression with polynomial basis, random forests, and K nearest neighbors. Simar, Halldor, and Kate will tackle each of them respectively.

3.1 Data

We plan to sample data from known functions/ distributions by adding noise and outliers to make it closer to the real-world setting. However there are couple of open questions. Should we use 2D or 3D data? Should we expose the sampling choice to the end-user or not? Should we allow user to express any arbitrary function/ distribution? How much noise/ outliers? Should the end-user make a decision? In the case of classification, how many classes? The exact procedure is an important design decision of our visualization system which we hope to decide after via deliberation.

3.2 K-nearest neighbors (KNN)

K Nearest Neighbors (KNN) is a well-established technique in classification based on training examples. For each unlabeled data point, we search for K closest training examples and label the data point as majority of neighbors.

Despite the intuitiveness of the KNN technique, its implementation requires to make a non-trivial choice of the value of K which, in turn, determines how successful the classification is. From the theory, we know that small values of K leads to high variance, and high values of K imply the high bias.

The self-explanatory story will illustrate what the bias and variance are in this context and how they are related to the choice of K. Moreover, there will be visual examples of possible advantages and pitfalls both for large and low values of K.

3.3 Linear models

Linear regression is a technique for modeling a dependent variable y as a linear combination of one or more independent variables x_i. Mathematically it can be expressed as $y_i = \sum(w_i x_i)$. Apart from capturing linear relationships, linear regression can also model non-linear relationships with a change in basis. One such basis is a polynomial basis, which can be expressed as a linear combination of independent variable x and its higher degree polynomials ($X = [1\, x x^2 x^3 \dots x^p]$).

The degree of polynomial basis p - to be determined by a practitioner - controls the complexity of the model. In this visualization, our task is to allow students to experiment with different values of p resulting in the different models in the bias-variance tradeoff landscape. We aim that the student will realize the fundamental tradeoff, relate it with the concept of overfitting/ underfitting and possibly decide on choosing optimal value of p.

3.4 Random Forests

Random forests [13] is an ensemble algorithm that is made up of many decision trees and outputs the mean or mode of those decision trees. A decision tree is a tree of optimal splitting rules to split a dataset based on its features as to segregate the classes of that dataset. Random forests try to reduce overfitting by fitting a decision tree to a number of bootstrap samples. Our task is to show how hyperparameters like max depth of the trees and number of trees affects the bias-variance tradeoff. By showing students how this change in parameters affects the model with dynamic visualisations and example classification tasks, we aim to give students an intuitive understanding of how these parameters affect the tradeoff.

4 Proposed solution

4.1 K-nearest neighbours (KNN)

For successful implementation of the KNN, a user wants to choose the optimal parameter K for the algorithm to get the least possible error. The user loads the website and is guided with self-explanatory tutorial to (consecutively)

1. Review how the KNN works. The user sees the figure that contains a background colored by true labels, a scatterplot of a few labeled points and of a few unlabeled points. The user can choose the value of K and see how the KNN assigns labels to unlabeled points as well as the percentage of the points labeled incorrectly (error). Clearly, the user aims to minimize this percentage. We suggest the user to do it empirically using visualization tools by choosing the value of K.
2. Understand what bias and variance are for this setting. The user is briefly introduced to the basic theory: the mean value of error corresponds to bias, variance is another important parameter of the error. The user is offered to run an experiment to estimate the bias and variance for different values of K. In the experiment, we want to emphasize that the bias and variance are computed over potential choices of labeled data. The exact viz solution is undecided yet. In addition, we explain what all combinations of low/high bias and low/high variance mean in a spirit of Fig. 1 in [11]
3. Visually understand how the bias changes as K increases. The user explores the interactive visualization where he can increase the value of K and see how more and more distant points are taken into the account, so the bias increases.
4. Visually understand how the variance changes as K increases. Recall that the variance of the error represents how far away from the bias (mean) the error is. To see the changes, we implement different strategy, namely, we see how the new data changes variance. Indeed, the mean of the error should not change if new data is available, but the variance may change. To illustrate this idea, the user will interact with the following figure. The user chooses the value of K. First, the user watches how the algorithm assigns values to the unlabeled data point based on the given labeled data points. The unlabeled data point is linked closest K labeled points. Then, by clicking the button, the user sees more a few more labeled points appear. Now we can see what happens with the error: for small values of K, there error decreases,

for large values of K, it remains intact.

5. Combine the knowledge into math formula. The viz solution is undecided yet. Depending on specific quantities that the programmer wants to optimize, this part if the tutorial will provide a visual guideline to motivate visually and mathematically certain choices of values of K.

4.2 Linear models

A student of machine learning learns about the parameter p in linear regression with polynomial basis. However, he is not clear on various aspects of the choice of p. He has many questions and wants to experiment with different values of the parameter, build machine learning models and figure out the predictability power of each model. Also, he wants to learn how to choose an optimal value of p by taking an informed decision on the bias and variance.

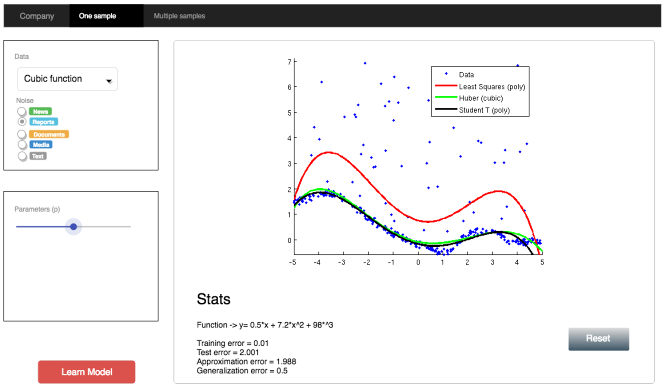
We hope to build an intuitive explanation of the above use-case where we will navigate the user through our interactive visualization. The explanation will introduce various machine-learning terminologies and concepts used in the analysis of choosing p. After the explanation, the user might have questions in his mind. We hand over the entire lab to the user where he tries various parameters of type of data distribution, values of p, noise, outliers and observes how the machine learning model behaves under different circumstances.

We hope to answer following questions/use-cases (we can rather call it learning-cases here) through this visualization.

1. What is bias/ variance tradeoff; how it is related to training/ approximation error; how does the concept of overfitting and underfitting relates to this?
2. How does the choice of p affects the bias/ variance tradeoff, training/ approximation error, overfitting/ underfitting?
3. How does the above change when we have noisy data/ outliers?

In Fig.1 and Fig.2, we present a tentative design of the interface and the interactive visualization. In Fig.1, various options give student the freedom to try different configurations of underlying data, noise, and parameter p. Based on these figures, the student will get the statistics like training error, testing error and generalization error. This will help him relate the model complexity with training/ testing error and hence to overfitting and underfitting.

In Fig.2, there are two possible designs and we are yet to choose which one to show. On the left, we learn multiple models with same value of p. The student observes the bias variance tradeoff for his chosen value of p. After he has observed for one value, he may want to compare how these curves changes with another value of p. On the right, we show values of bias and variance with different values of p on the same curve. The former is more intuitive but it requires user to remember previous curves, putting cognitive load on the user.



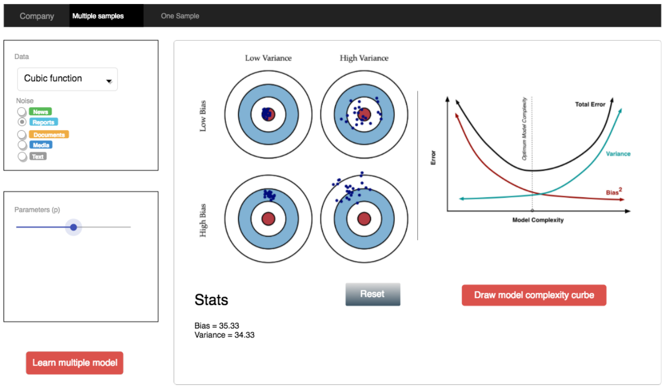
Cubic function

Stats

Function -> y= 0.5*x + 7.2*x^2 + 98*^3

Training error = 0.01
Test error = 2.001
Approximation error = 1.988
Generalization error = 0.5

Fig. 1. Visualization 1: Data is shown with blue dots. Different models corresponding to different values of p are shown by lines of different colors. The user has a choice of selecting the value of p with a slider and he can learn a model using the button. This allows the student visually see how different models fit the training data.



Cubic function

Stats

Bias = 35.33
Variance = 34.33

Fig. 2. Visualization 2: There are two proposed solutions. On the left, the student observes the bias-variance tradeoff for one single value of p at a time. On the right, he observes bias and variance with different values of p on the same curve.

The exact design will be finalized with further analysis and deliberation which will take into account various resources like cognitive load, attention span, external and internal memory of the end-users. We will ponder over design-choices like faceting, animation, filtering, aggregation, summarization, etc., while focusing on the principle of effectiveness.

4.3 Random Forests

Many students struggle with the concept of the bias-variance tradeoff, especially how it manifests itself in the Random Forest classification algorithm which students also often find hard to grasp. In this part of our project the student loads up the webpage and is greeted with a series of visualizations embedded within text that together give a cohesive story on the bias-variance tradeoff with respect to Random Forests.

We start by showing a visualization of a single decision tree. The figure is a scatterplot of two features with the decision boundaries of the decision trees drawn (See Figure 3.

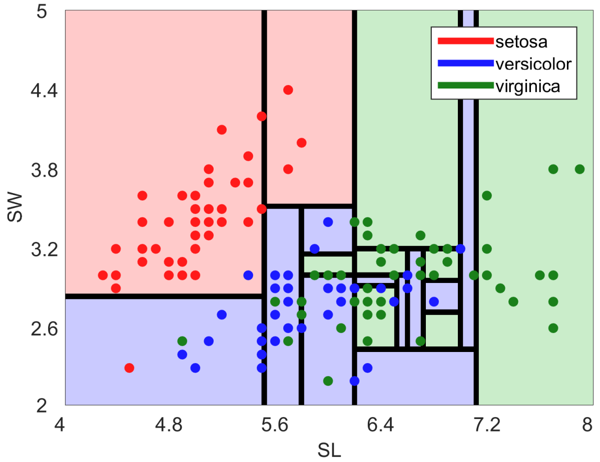


Fig. 3. A visualisation of a decision tree's boundaries for two features. From [14].

There are also a few test points plotted on the scatterplot. We allow the user to play around with the max-depth of the decision tree, allowing the user to see how it changes the bias and the variance of the model. On the left side of the plot would be the actual tree representation of the decision tree. On to the right side there is a graph similar to a heart monitor showing how the test error and the training error change as the student adjusts the hyperparameters. We then try to engage the student by coming up with a way to lower the variance of the model. What if we generated k many trees and took the average of them? Since it is a deterministic algorithm, not much would happen.

Then we show the students how to solve this with bootstrap sampling. Here we could show an optional explanatory figure for sampling with replacement. We then show how generating more trees and taking a majority vote for each datapoint provides an intuitive way of visualizing how overfitting is reduced. Here, were not completely set on the type of idiom but something along the lines of the R2D3 visualisation [15] (see Figure 4). The emphasis is on how a datapoint flows through the decision tree. However, our visualization will have multiple decision trees and a very intuitive way of explaining the voting process. One idea is to put a graph of the test error on the right side, and see how the test error goes down when more trees are added.

5 Implementation approach

Our finished project will be a webpage encompassing all our visualizations, explanatory text and formulas. We will use **D3.js** as our implementation approach for the visualizations. We chose D3.js because we know its the state of the art visualization framework so we all want to learn and become proficient in using it. We also know D3.js is very flexible so if we get more visualization ideas as we go along we know they can be implemented with D3.js. The text will styled and rendered using **Markdown**. We have not decided yet if we will use scrolling as a way to make our visualizations dynamic. On one hand it adds flavor to the explanation but on the other hand we feel that it's hard to add interactivity.

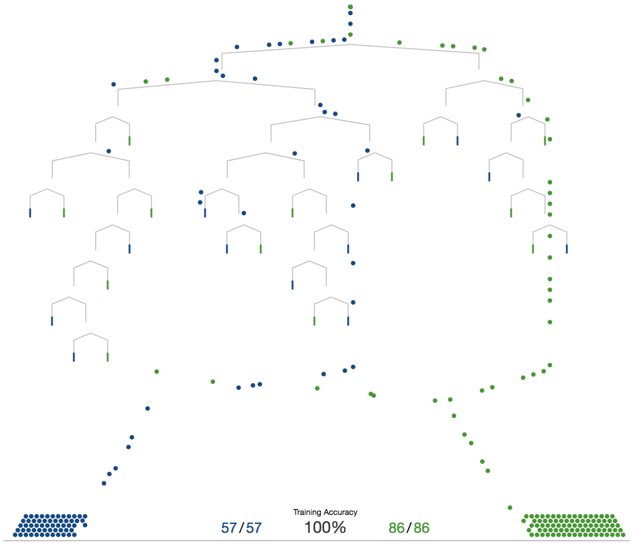


Fig. 4. A visualisation of a decision tree "in-action". We propose to do something similar for Random Forests. From [15].

6 Milestones

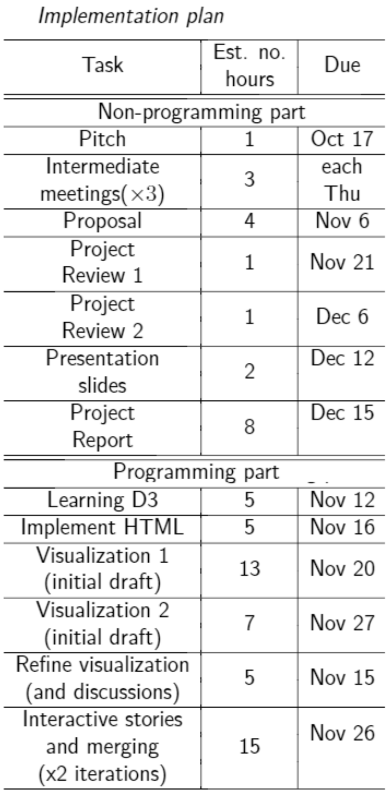
We plan to start by learning and getting acquainted with the various technologies like HTML, CSS and mainly D3. Each of the team member is working on a separate algorithm to showcase the bias-variance tradeoff. Fig.5 describes a detailed breakout of time spent by each team member.

7 Previous work

There is a good amount of work in this domain [10, 11, 12]. However, many of these are focused on writing good text and adding figures only to complement the text. This approach is not sufficient when it comes to interactivity and intuitiveness.

For instance, the relationship between KNN's hyperparameter K and the bias-variance tradeoff has been explained in great detail by Scott Fortmann [16] in his online article. However, this was done with visualizations that had very limited interactivity. Similarly, [17] provides an intuitive explanation on the interpretation of RMSE loss for linear regression. However, this work does not reflect on the bias-variance tradeoff.

Also, most visualizations covering random forests are simple static visualizations that are not very intuitive. A very nice intuitive and animated visualization story has been made for decision trees by Stephanie Yee and Tony Chu [15]. However, this has not been extended to cover Random Forests and how the bias-variance tradeoff affects decision trees. Another popular example on the web [18] shows random forests on a high level with dynamic visualizations. However, we think its more intuitive to show how an individual datapoints



Implementation plan

Task	Est. no. hours	Due
Non-programming part		
Pitch	1	Oct 17
Intermediate meetings($\times 3$)	3	each Thu
Proposal	4	Nov 6
Project Review 1	1	Nov 21
Project Review 2	1	Dec 6
Presentation slides	2	Dec 12
Project Report	8	Dec 15
Programming part		
Learning D3	5	Nov 12
Implement HTML	5	Nov 16
Visualization 1 (initial draft)	13	Nov 20
Visualization 2 (initial draft)	7	Nov 27
Refine visualization (and discussions)	5	Nov 15
Interactive stories and merging (x2 iterations)	15	Nov 26

Fig. 5.

flows through the decision trees and then show how a majority vote is used to output the classification. Our version therefore combines the strenghts of the two methods listed above.

However, [19] provides an awesome tool to tinker various parameters of neural network. We wish to study this in detail, take inspiration to implement this for bias-variance tradeoff.

References

[1] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al., 2011. "The kaldi speech recognition toolkit". In IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584, IEEE Signal Processing Society.

[2] Graves, A., Jaitly, N., and Mohamed, A.-r., 2013. "Hybrid speech recognition with deep bidirectional lstm". In Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, IEEE, pp. 273–278.

[3] LeCun, Y., Bengio, Y., and Hinton, G., 2015. "Deep learning". *Nature,* **521**(7553), pp. 436–444.

[4] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L., 2016. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". *arXiv preprint arXiv:1606.00915*.

[5] Sarikaya, R., Hinton, G. E., and Deoras, A., 2014. "Application of deep belief networks for natural language understanding". *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP),* **22**(4), pp. 778–784.

[6] Sutskever, I., Vinyals, O., and Le, Q. V., 2014. "Sequence to sequence learning with neural networks". In Advances in neural information processing systems, pp. 3104–3112.

[7] Ros, G., Sappa, A., Ponsa, D., and Lopez, A. M., 2012. "Visual slam for driverless cars: A brief survey". In Intelligent Vehicles Symposium (IV) Workshops, Vol. 2.

[8] Insider, B., 2017. There's a shortage of ai engineers in the us.

[9] Pappano, L., 2012. "The year of the mooc". *The New York Times,* **2**(12), p. 2012.

[10] Fortmann, S., 2012. Understanding the bias-variance tradeoff.

[11] Stephanie, and Tony, 2017. A visual introduction to machine learning.

[12] Harris, N., 2014. Visualizing k-means clustering.

[13] Breiman, L., 2001. "Random forests". *Machine learning,* **45**(1), pp. 5–32.

[14] Eicholtz, M. viewboundary.

[15] Chu, S. Y. . T., 2015. A visual introduction to machine learning.

[16] Booklet, A., 1994. Booklet title. On the WWW, at `http://www.abc.edu`, May. PDF file.

[17] Visually, E., 2014. Ordinary least squares regression.

[18] Chen, S. M. . Y., 2017. A path to random forest.

[19] Flow, T., 2015. Tinker with a neural network.