Lecture 4: k-nearest neighbours and SVM RBFs

Firas Moosvi (Slides adapted from Varada Kolhatkar)

UBC Computer Science

Announcements

- My office hours are Wednesdays from 1:30 - 2:30 PM in ICCS 253.
- hw2 was due yesterday.
- Syllabus quiz due date is September 19th, 11:59 pm.
- Homework 3 (hw3) has been released (Due: Oct 1st, 11:59 pm)
    - You can work in pairs for this assignment.
- If you were on the waitlist, you should know your enrollment status now. Attendance in tutorials is not mandatory; they are optional and will follow an office-hour format. You are free to attend any tutorial session of your choice.
- The lecture notes within these notebooks align with the content presented in the videos. Even though we do not cover all the content from these notebooks during lectures, it's your responsibility to go through them on your own.

Class 3 Demo Continued

For the first 15-20 mins, we’ll try to finish off the demo from Lecture 3.

You can find the demo repo here, though you have probably created a repo already locally from Thursday.

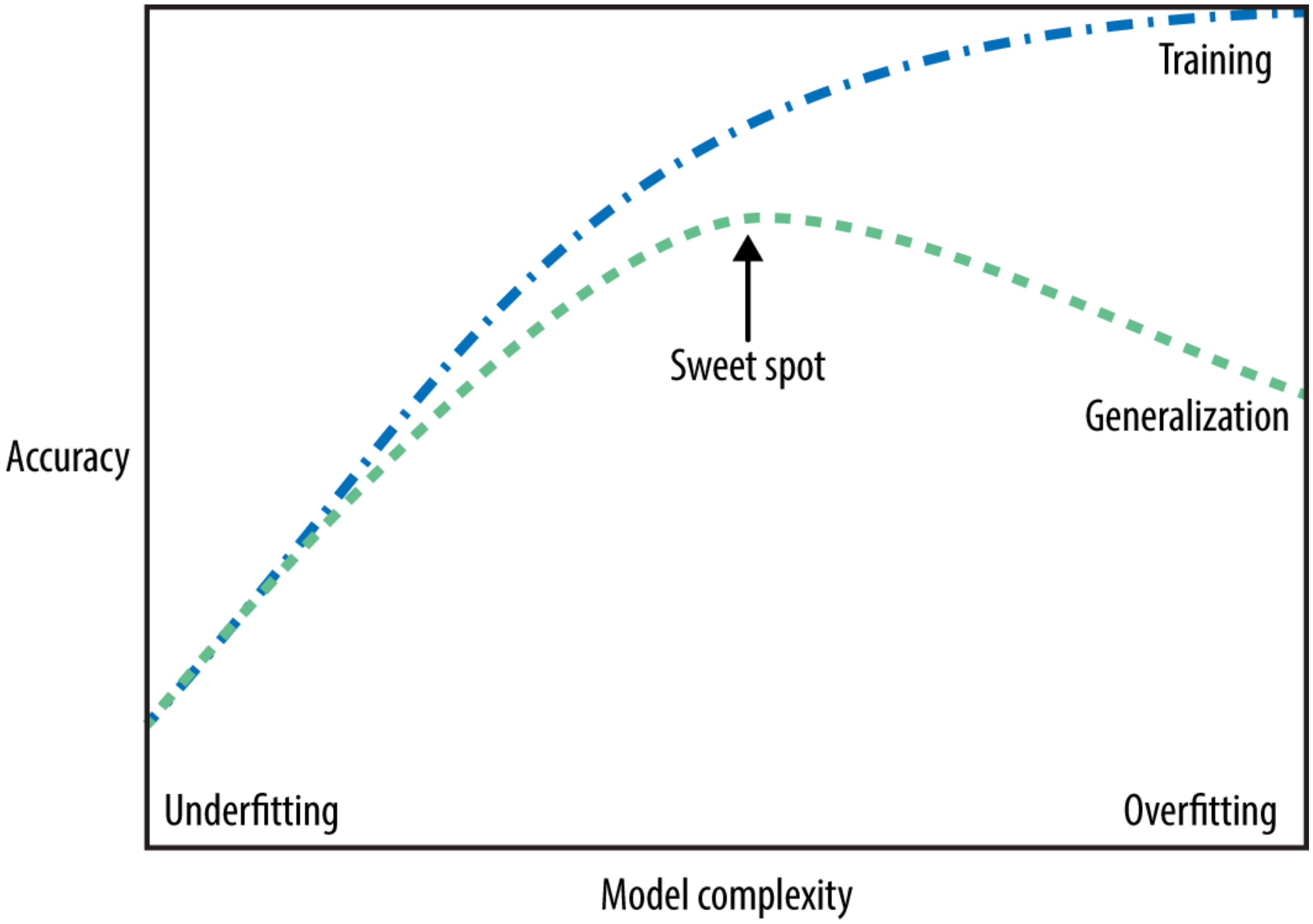
You can find the finished version of the repo in my demo account after class.

Pod Work: Discuss these questions

- Why do we split data?
- What are train/valid/test splits?
- What are the benefits of cross-validation?
- What's the fundamental trade-off in supervised machine learning?
- What is the golden rule of machine learning?

Recap: The fundamental tradeoff

As you increase the model complexity, training score tends to go up and the gap between train and validation scores tends to go up.



Pod Work: Discuss this question

Which of the following statements about **overfitting** is true?

- a. Overfitting is always beneficial for model performance on unseen data.
- b. Some degree of overfitting is common in most real-world problems.
- c. Overfitting ensures the model will perform well in real-world scenarios.
- d. Overfitting occurs when the model learns the training data too closely, including its noise and outliers.

Pod Work: Discuss this question

Which of the following scenarios do **NOT necessarily imply overfitting**?

- a. Training accuracy is 0.98 while validation accuracy is 0.60.
- b. The model is too specific to the training data.
- c. The decision boundary of a classifier is wiggly and highly irregular.
- d. Training and validation accuracies are both approximately 0.88.

Pod Work: Discuss this question

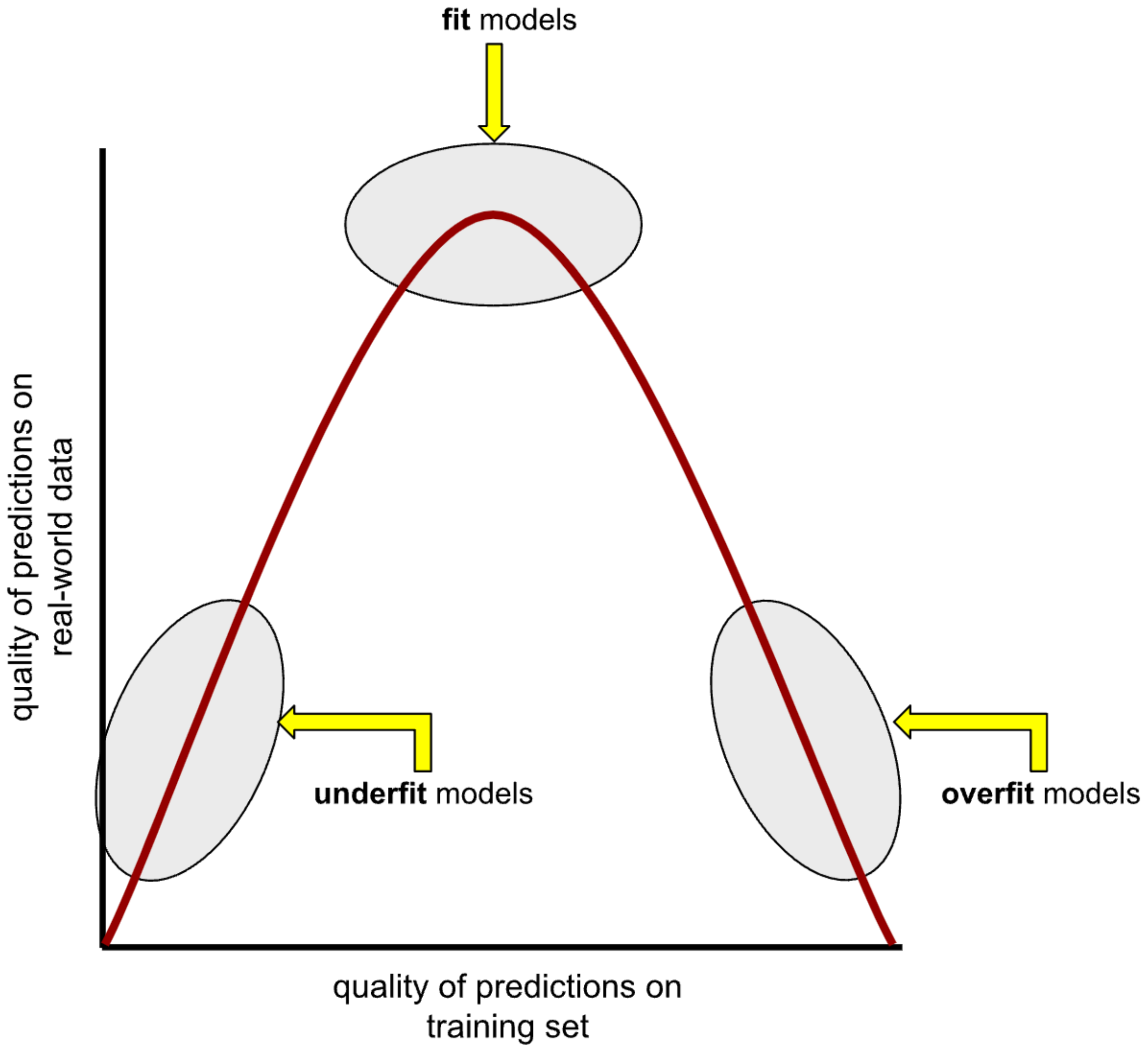
How might one address the issue of **underfitting** in a machine learning model.

- a. Introduce more noise to the training data.
- b. Remove features that might be relevant to the prediction.
- c. Increase the model’s complexity, possibly by adding more parameter or features
- d. Use a smaller dataset for training.
- e. Use a larger dataset for training.

Overfitting and underfitting

- An **overfit model** matches the training set so closely that it fails to make correct predictions on new unseen data.
- An **underfit model** is too simple and does not even make good predictions on the training data

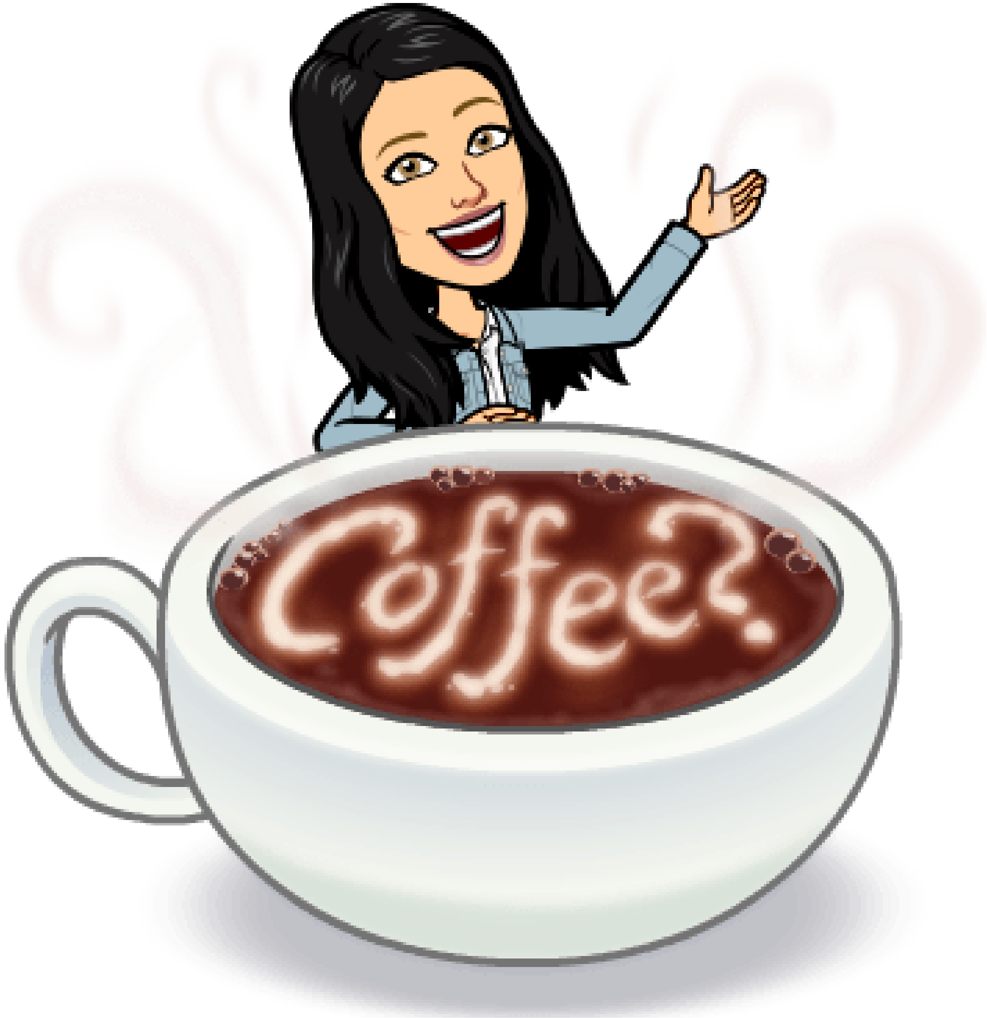
Overfitting and underfitting


fit models
underfit models
overfit models
quality of predictions on real-world data
quality of predictions on training set

Source

Break

Let's take a break!



iClicker 4.1

iClicker cloud join link: https://join.iclicker.com/VYFJ

Select all of the following statements which are TRUE.

- a. Analogy-based models find examples from the test set that are most similar to the query example we are predicting.
- b. Euclidean distance will always have a non-negative value.
- c. With k-NN, setting the hyperparameter k to larger values typically reduces training error.
- d. Similar to decision trees, k-NNs finds a small set of good features.
- e. In k-NN, with $k > 1$, the classification of the closest neighbour to the test example always contributes the most to the prediction.

iClicker 4.2

iClicker cloud join link: https://join.iclicker.com/VYFJ

Select all of the following statements which are TRUE.

- a. k-NN may perform poorly in high-dimensional space (say, $d > 1000$).
- b. In sklearn’s SVC classifier, large values of `gamma` tend to result in higher training score but probably lower validation score.
- c. If we increase both `gamma` and `C`, we can’t be certain if the model becomes more complex or less complex.

Similarity-based algorithms

- Use similarity or distance metrics to predict targets.
- Examples: k-nearest neighbors, Support Vector Machines (SVMs) with RBF Kernel.

k-nearest neighbours

- Classifies an object based on the majority label among its k closest neighbors.
- Main hyperparameter: k or `n_neighbors` in `sklearn`
- Distance Metrics: Euclidean
- Strengths: simple and intuitive, can learn complex decision boundaries
- Challenges: Sensitive to the choice of distance metric and **scaling** (coming up).