Lecture 4: k-nearest neighbours and SVM RBFs

Firas Moosvi (Slides adapted from Varada Kolhatkar)

Announcements

- Reminder: hw2 is due January 19th!
- Add/Drop date is tomorrow!

Situation in Iran

- Come talk to me if there's something you think I can do.
- Science Embedded Counsellor and UBC for other supports
- Support for unanticipated circumstances

Finish up Lecture 3 Demo

For this demo, each student should click this link to create a new repo in their accounts, then clone that repo locally to follow along with the demo from today.

Recap: Clicker 4.0a

Participate using Agora (code: agentic)

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

- a. Training accuracy is very high (0.98) while validation accuracy is much lower (0.60).
- b. In a wildlife classifier, the model predicts “wolf” whenever there”s snow in the background, because all wolf photos were taken in snowy regions.
- c. The decision boundary of a classifier is wiggly and highly irregular.
- d. Training and validation accuracies are both approximately 0.88.
- e. A cancer detection model learns that “a ruler in the corner of the X-ray” means positive, because doctors tended to measure suspicious cases.

Recap: Clicker 4.0b

Participate using Agora (code: agentic)

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

- a. Overfitting makes the model more accurate on both training and unseen data.
- b. Overfitting means the model captures noise or irrelevant details from the training data.
- c. Overfitting is desirable because it reduces both training and test error.
- d. In real-world problems, models are always at risk of overfitting if not properly validated.

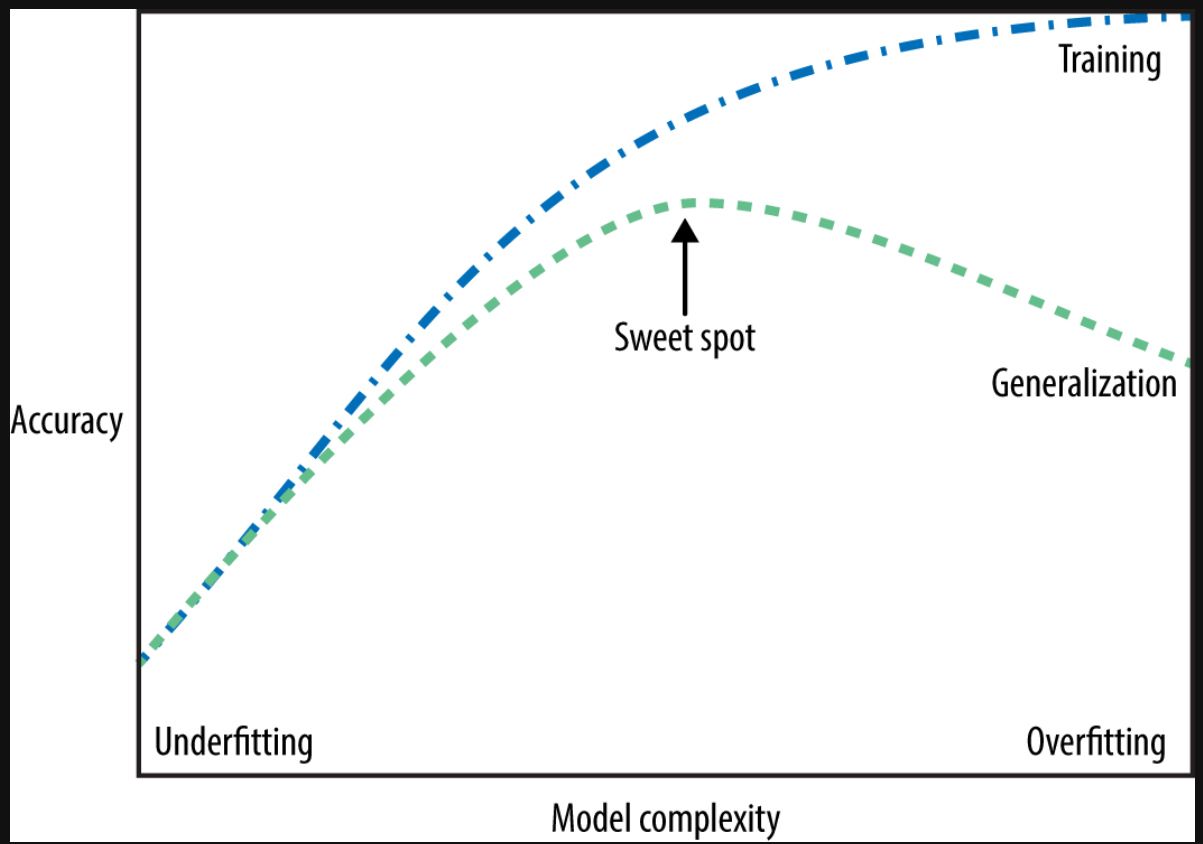
Recap: Clicker 4.0c

Participate using Agora (code: agentic)

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 (e.g., more parameters, features, or deeper trees)
- d. Use a smaller dataset for training.

The fundamental tradeoff

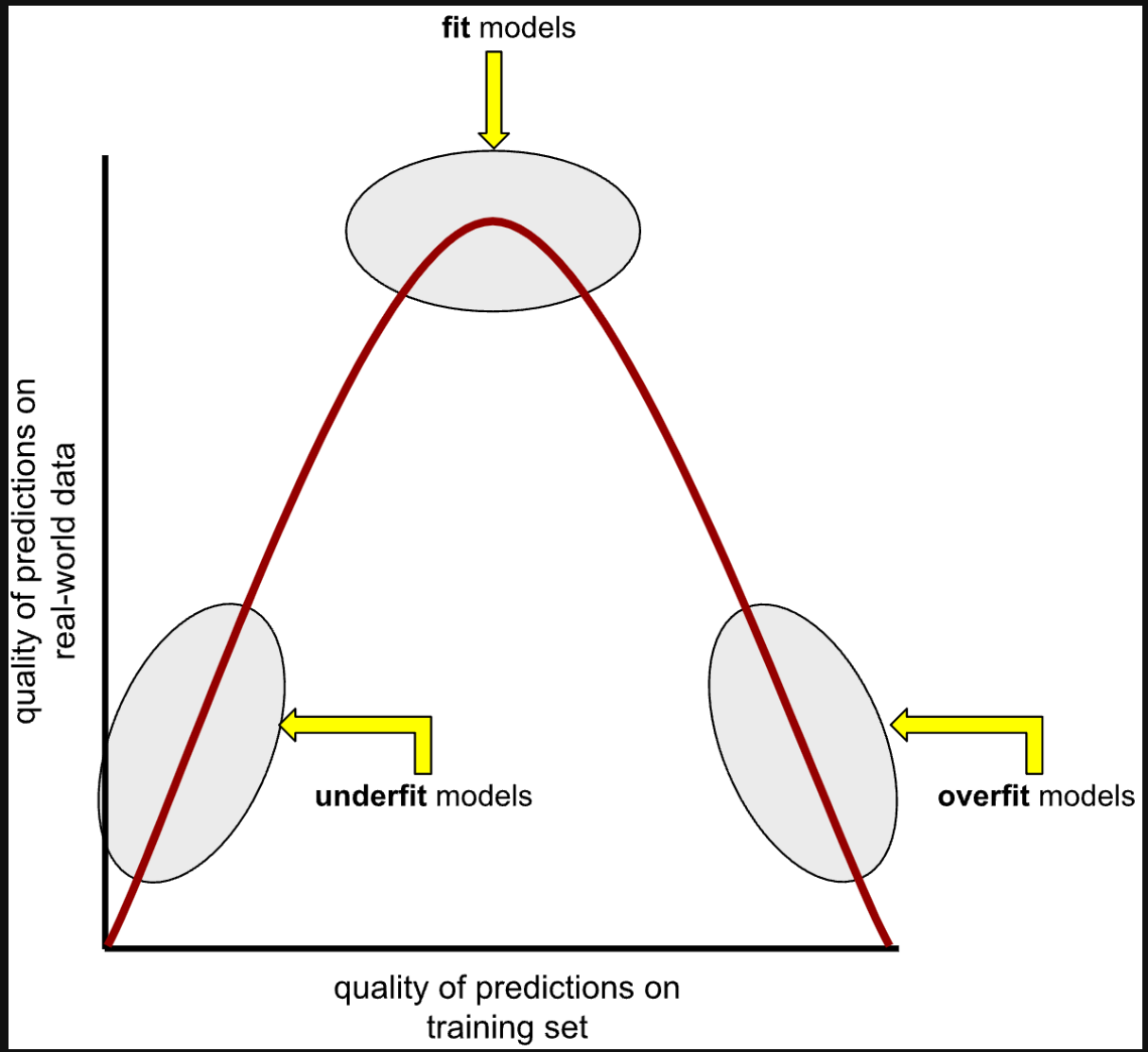


- As you increase the model complexity, training score tends to go up and the gap between train and validation scores tends to go up.
- How to pick a model?

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



Source

Clicker 4.1

Participate using Agora (code: agentic)

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.

Clicker 4.2

Participate using Agora (code: agentic)

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).

Curse of dimensionality

- As dimensionality increases, the volume of the space increases exponentially, making the data sparse.
- Distance metrics lose meaning
  - Accidental similarity swamps out meaningful similarity
  - All points become almost equidistant.
- Overfitting becomes likely: Harder to generalize with high-dimensional data.
- How to deal with this?
  - Dimensionality reduction (PCA) (not covered in this course)
  - Feature selection techniques.

SVMs with RBF kernel

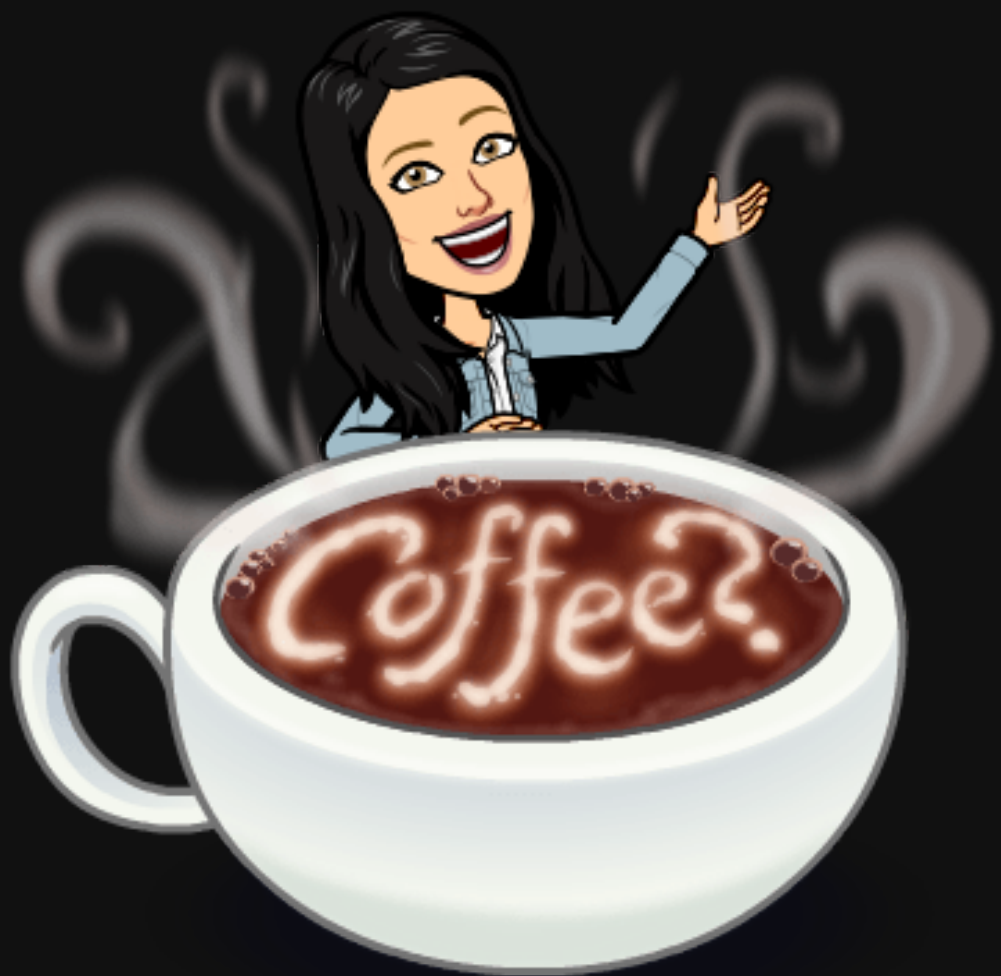
- RBF Kernel: Radial Basis Function, a way to transform data into higher dimensions implicitly.
- Strengths
  - Effective in high-dimensional and sparse data
  - Good performance on non-linear problems.
- Hyperparameters:
  - C: Regularization parameter (trade-off between correct classification of training examples and maximization of the decision margin).
  - gamma (γ): controls how fast the similarity decays with distance

Intuition of `C` and `gamma` in SVM RBF

- `C` (Regularization): Controls the trade-off between perfect training accuracy and having a simpler decision boundary.
  - High C: Strict, complex boundary (overfitting risk).
  - Low C: More errors allowed, smoother boundary (generalizes better).
- `Gamma` (Kernel Width): Controls the influence of individual data points.
  - High Gamma: Points have local impact, complex boundary.
  - Low Gamma: Points affect broader areas, smoother boundary.
- Key trade-off: Proper balance between `C` and `gamma` is crucial for avoiding overfitting or underfitting.

Break

Let’s take a break!



Group Work: Class Demo & Live Coding

There is no lecture demo for today because it requires some complex setup - just watch the instructor demo it live!

Models

Supervised models we have seen

- Decision trees: Split data into subsets based on feature values to create decision rules
- k-NNs: Classify based on the majority vote from k nearest neighbors
- SVM RBFs: Create a boundary using an RBF kernel to separate classes

Comparison of models (activity)

Model	Parameters and hyperparameters	Strengths	Weaknesses
Decision Trees			
KNNs			
SVM RBF			