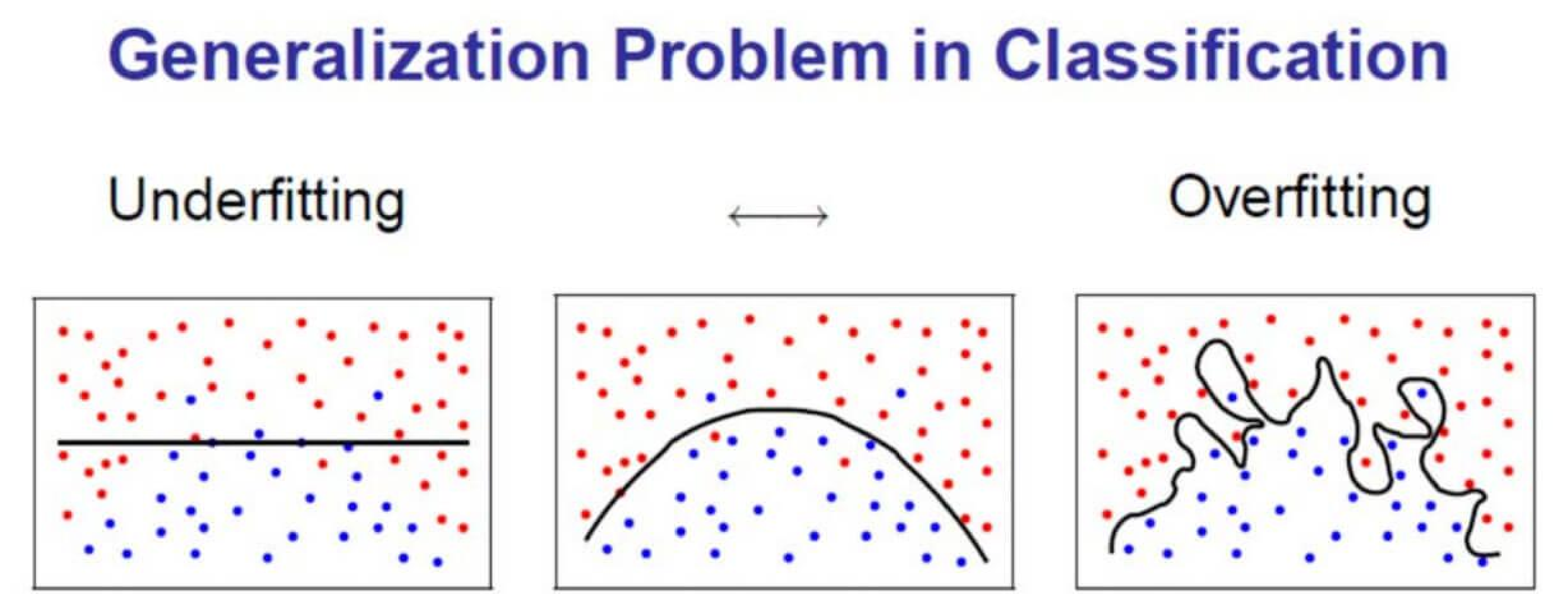
Random Forest Classifiers

Machine Learning with Networking Flow Data

The aim(s) of any ML model

- Should **GENERALIZE** well on the data
- Should not **OVERFIT** the data
- Should not **UNDERFIT** the data

Illustrating Overfitting, Underfitting and "Just right"



The problem with decision trees : **overfitting is easy**

- Decision trees are able to come up with complex models which generally tend to **overfit** on the data.

- A good way to imagine how this happens is to recall the extremely complicated flow chart that was classifying iris species from the last session.

- Since decision trees tend to overfit they fail to generalize well.

Solution

Use **Ensemble methods** like **Random Forest Classifiers** or **Gradient boosted regression trees**

Ensemble methods can be thought of as methods that combine a number of similar models to reduce overfitting.

Random Forest Classifiers

- Since decision trees tend to overfit the data - we use multiple decision trees.
- Although this might sound counterintuitive, if we have multiple trees all of which are overfitting, our model generalizes well.
- I like to think of this as a multi party democracy (like India) : If every political party or group has an equal say, one party cannot get away with getting their interests fulfilled as everyone is vying for their interests to be met.
- Analogously, none of the decision trees’ overfitting matters and their effects cancel out.
- This generalization can also be shown using rigorous mathematics.

Application to network flow data

The NIMS dataset

- Consists of packets internally collected at the University of Dalhousie research testbed.
- Different network scenarios emulated using computers to capture network traffic.
- The flows are obtained/observed using a tool called NetMATE.
- The labeled traffic is classified into multiple classes - **Remote connection, DNS, SCP, HTTP, FTP, P2P, TelNet,** etc.
- Thus, our goal is to model this multi-class classification problem using Random Forest Classifiers and then predict newer incoming traffic into one of the aforementioned labeled classes.
- The NIMS data contains 713,851 rows and 23 columns.

The NIMS dataset contd. (by the Univ. of Dalhousie, Canada)

FEATURES:

Protocol (proto)	Duration of the flow (Duration)
# Packets in forward direction (fpackets)	# Bytes in forward direction (fbytes)
# Packets in backward direction (bpackts)	# Bytes in backward direction (bbytes)
Min forward inter-arrival time (min_fiat)	Min backward inter-arrival time (min_biat)
Std deviation of forward inter-arrival times (std_fiat)	Std deviation of backward inter-arrival times (std_biat)
Mean forward inter-arrival time (mean_fiat)	Mean backward inter-arrival time (mean_biat)
Max forward inter-arrival time (max_fiat)	Max backward inter-arrival time (max_biat)
Min forward packet length (min_fpkt)	Min backward packet length (min_bpkt)
Max forward packet length (max_fpkt)	Max backward packet length (max_bpkt)
Std deviation of forward packet length (std_fpkt)	Std deviation of backward packet length (std_bpkt)
Mean backward packet length (mean_fpkt)	Mean forward packet length (mean_bpkt)

Implementation

Using Scikit-learn

99.89% - TESTING

99.9% - TRAINING

Therefore, no OVERFITTING!

THANK YOU!

...