Machine Learning

CSE 4308/5360: Artificial Intelligence I

University of Texas at Arlington

Machine Learning

- Machine learning is useful for constructing agents that improve themselves using observations.
- Instead of hardcoding how the agent to behave, we allow the behavior to be optimized based on training data.
- In many AI applications in speech recognition, computer vision, game-playing, etc., machine learning methods vastly outperform hardcoded agents.

Pattern Recognition

- In pattern recognition (aka pattern classification) the setting is this:
- We have patterns, which can be, for example:
  - Images or videos.
  - Strings.
  - Sequences of numbers, booleans, or strings (or a mixture thereof).
- We have classes, and each pattern is associated with a class.

Pattern	Class
A photograph of a face	The human
A video of a sign from American Sign Language	The sign
A book (represented as a string)	The genre of the book.

- Our goal: build a system that, given a pattern, estimates its class.
  - E.g., given a photograph of a face, recognize a person.
  - Given a video of a sign, recognize the sign.

Pattern Recognition

- More formally: the goal in pattern recognition is to construct a **classifier** that is as accurate as possible.
- A classifier is a function F, mapping patterns to classes.
  F: set of patterns → set of classes.
  - The input to F is a pattern (e.g., a photograph of a face).
  - The output of F is a class (the ID of the human that the face belongs to).
- Typically, classifiers are not perfect.
  - In most real-world cases, the classifier will make some mistakes, and for some patterns it will output the wrong class.
- One key measure of performance of a classifier is its **error rate**: the percentage of patterns for which F provides the wrong answer.
  - Obviously, we want the error rate to be as low as possible.
- Another term is **classification accuracy**, equal to 1 – error rate.

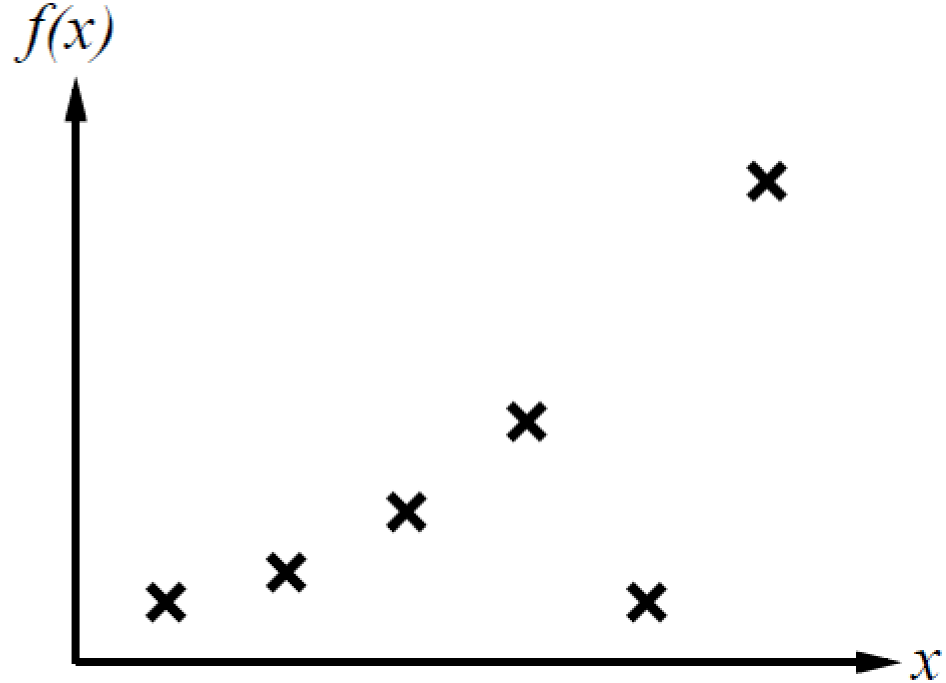
Learning and Recognition

- Machine learning and pattern recognition are **<u>not</u>** the same thing.
  - This is a point that confuses many people.
- You can use machine learning to learn things that are not classifiers. For example:
  - Learn how to walk on two feet.
  - Learn how to grasp a medical tool.
- You can construct classifiers without machine learning.
  - You can hardcode a bunch of rules that the classifier applies to each pattern in order to estimate its class.
- However, machine learning and pattern recognition are heavily related.
  - A big part of machine learning research focuses on pattern recognition.
  - Modern pattern recognition systems are usually exclusively based on machine learning.

Supervised Learning

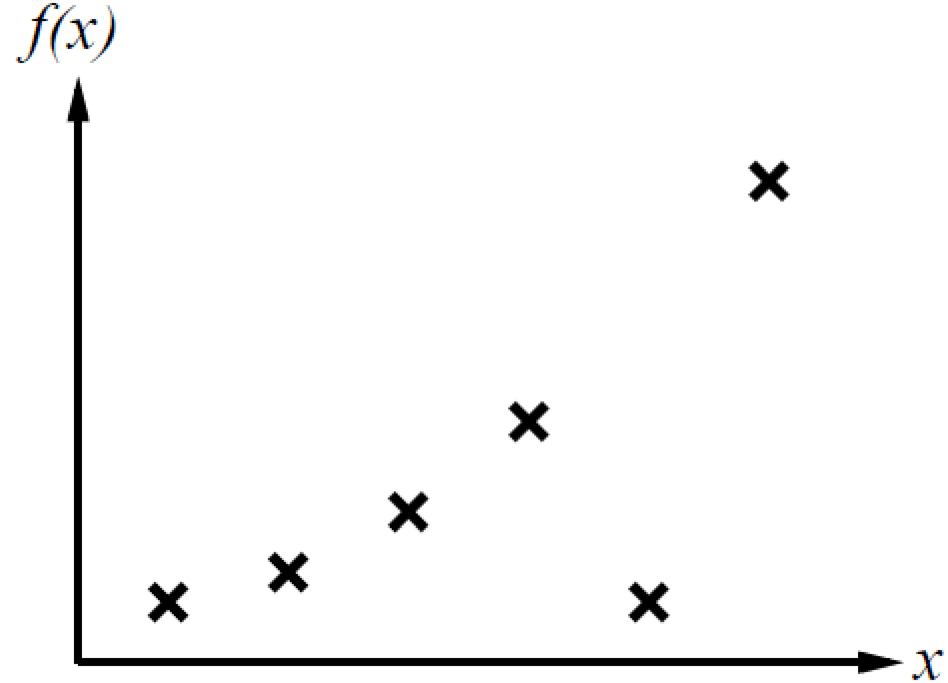
- In supervised learning, our training data is a set of pairs.
- Each pair consists of:
  - A pattern.
  - The true class for that pattern.
- Another way to think about this is this:
  - There exists a perfect classifier F_{true}, that knows the true class of each pattern.
  - The training data gives us the value of F_{true} for many examples.
  - Our goal is to learn a classifier F, mapping patterns to classes, that agrees with F_{true} as much as possible.
- The difficulty of the problem is this:
  - The training data provide values of F_{true} for only some patterns.
  - Based on those examples, we need to construct a classifier F that provides an answer for ANY possible pattern.

Supervised Learning Example



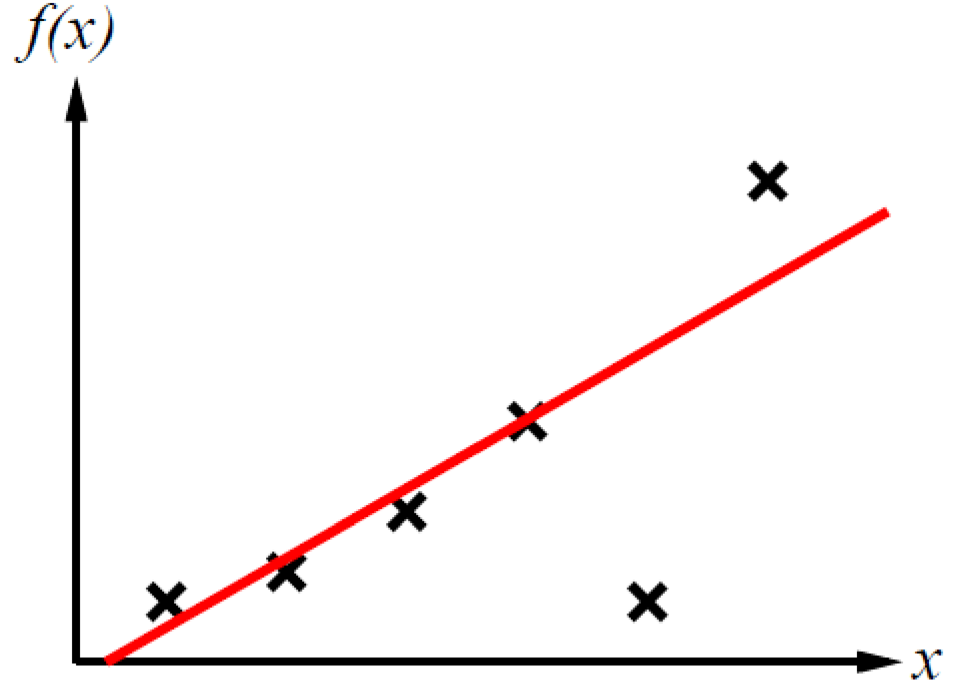
- This is a toy example.
  - From the textbook.
- Here, the “pattern” is a single real number.
- The class is also a real number.
- So, F_{true} is a function from the reals to the reals.
  - Usually patterns are much more complex.
  - In this toy example it is easy to visualize training examples and classifiers.
- Each training example is an X on the figure.
  - The x coordinate is the pattern, the y coordinate is the class.
- Based on these examples, what do you think F_{true} looks like?

Supervised Learning Example



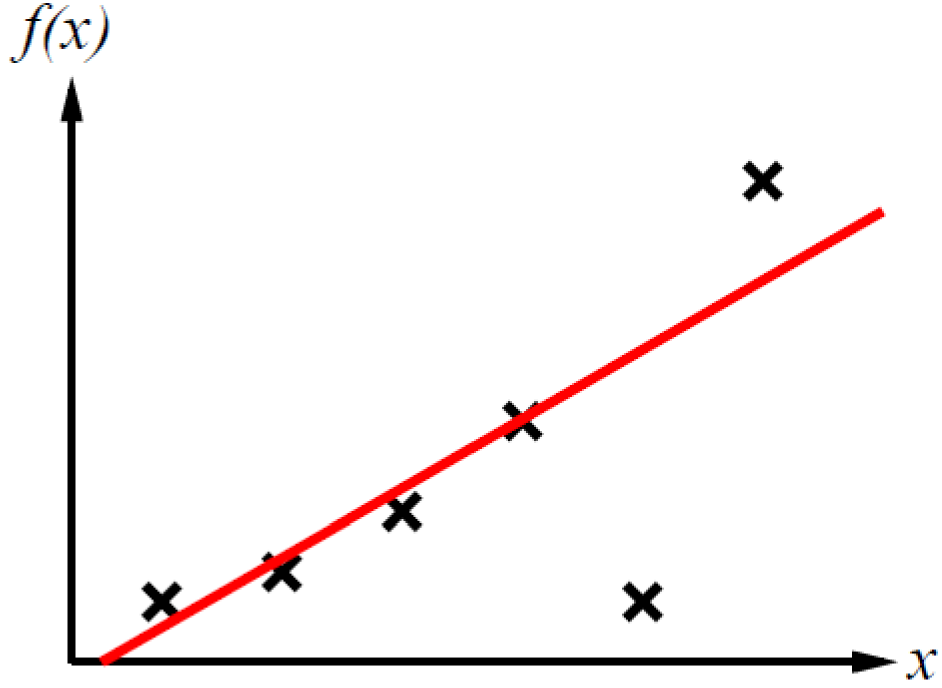
- Different people may give different answers as to what F_{true} may look like.
- That shows the challenge in supervised learning: we can find some plausible functions, but how do we know that one of them is correct?

Supervised Learning Example



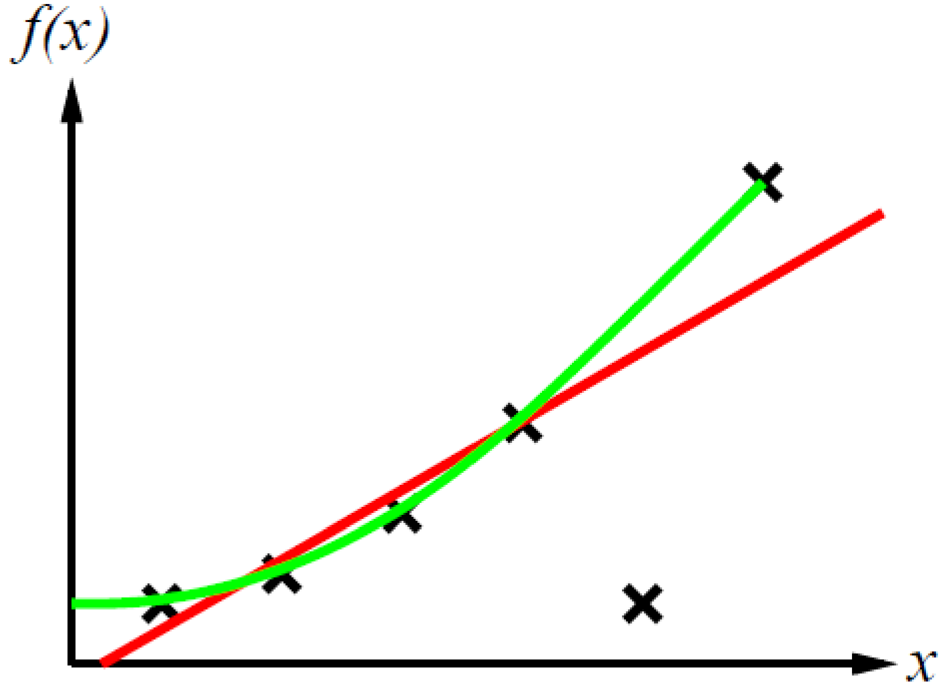
- Here is one possible classifier F.
- Can anyone guess how it was obtained?

Supervised Learning Example



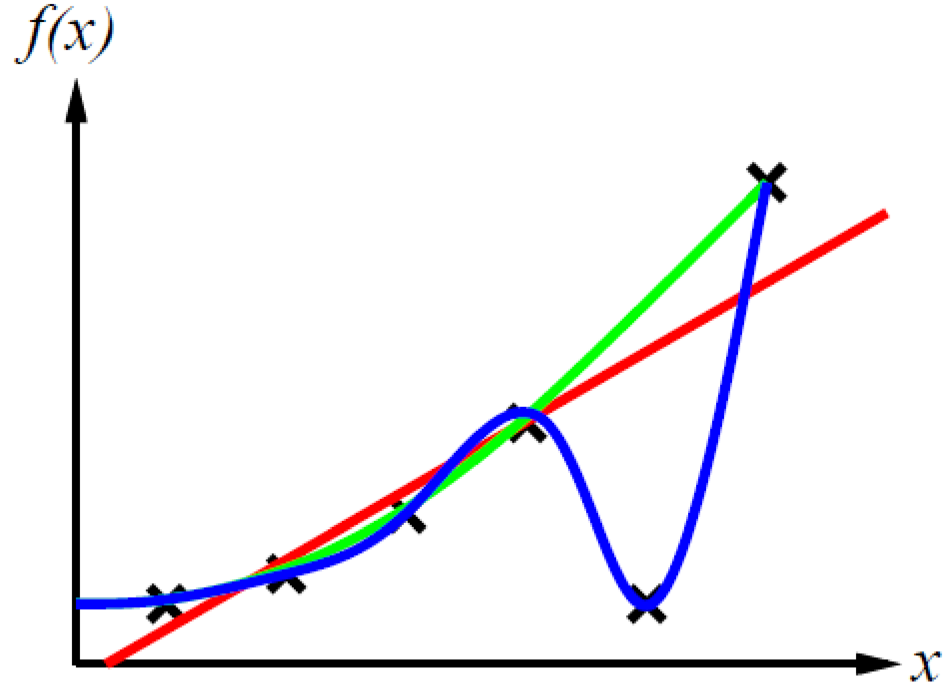
- Here is one possible classifier F.
- Can anyone guess how it was obtained?
- It was obtained by fitting a line to the training data.

Supervised Learning Example



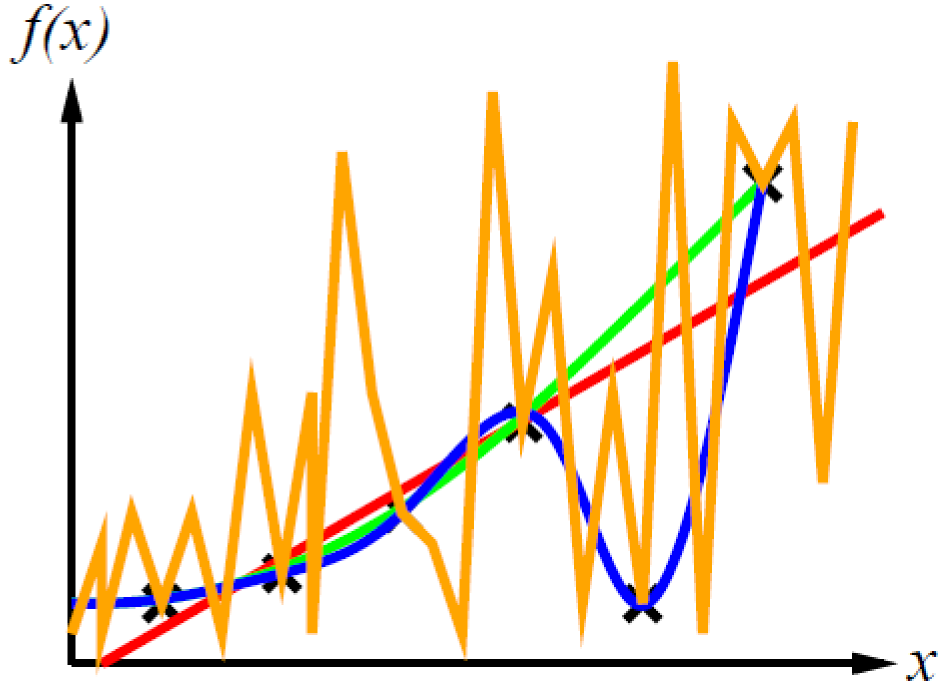
- Here we see another possible classifier F, shown in green.
- It looks like a quadratic function (second degree polynomial).
- It fits all the data perfectly, except for one.

Supervised Learning Example



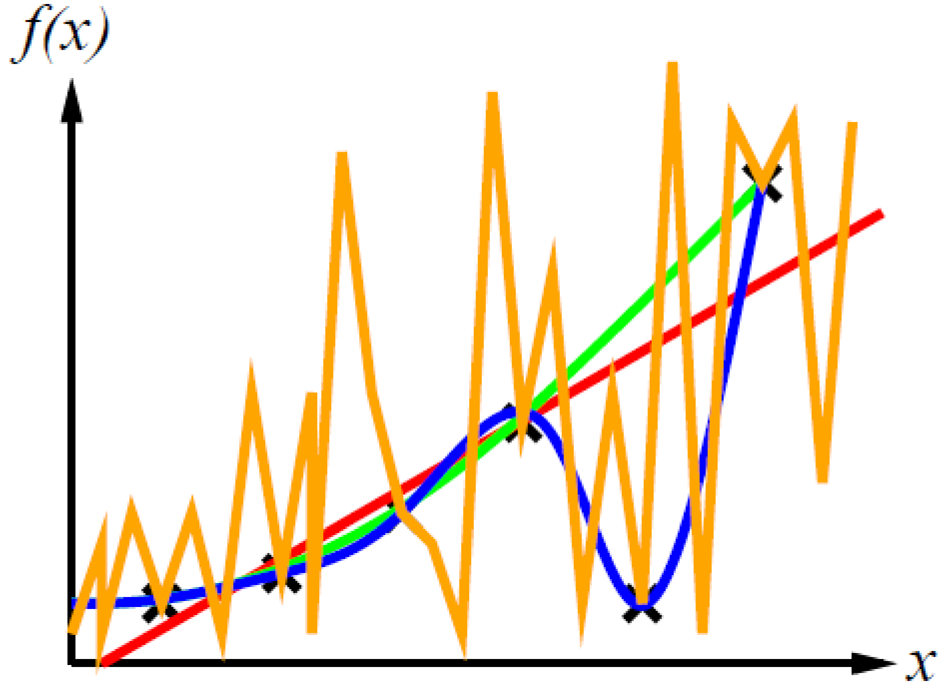
- Here we see a third possible classifier F, shown in blue.
- It looks like a cubic degree polynomial.
- It fits all the data perfectly.

Supervised Learning Example



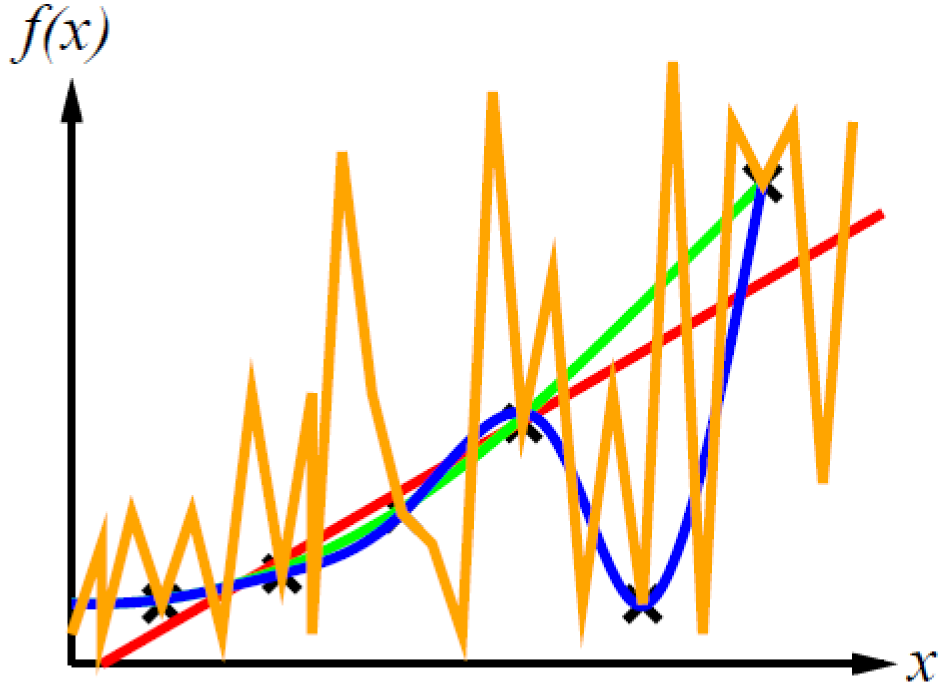
- Here we see a fourth possible classifier F, shown in orange.
- It zig-zags a lot.
- It fits all the data perfectly.

Supervised Learning Example



- Overall, we can come up with an infinite number of possible classifiers here.
- The question is, how do we choose which one is best?
- Or, an easier version, how do we choose a good one.
- Or, an easier version: given a classifier, how can we measure how good it is?
- **What are your thoughts on this?**

Supervised Learning Example

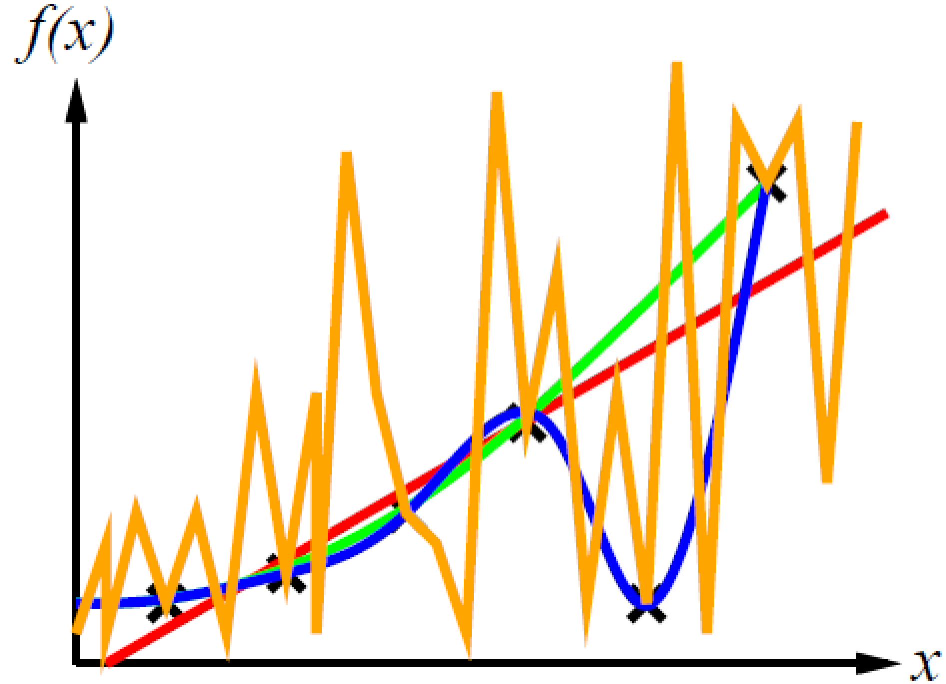


- One naïve solution is to evaluate classifiers based on **training error**.
- For any classifier F, its training error can be measured as a sum of squared errors over training patterns X:

$$\sum_X [F_{true}(X) - F(X)]^2$$

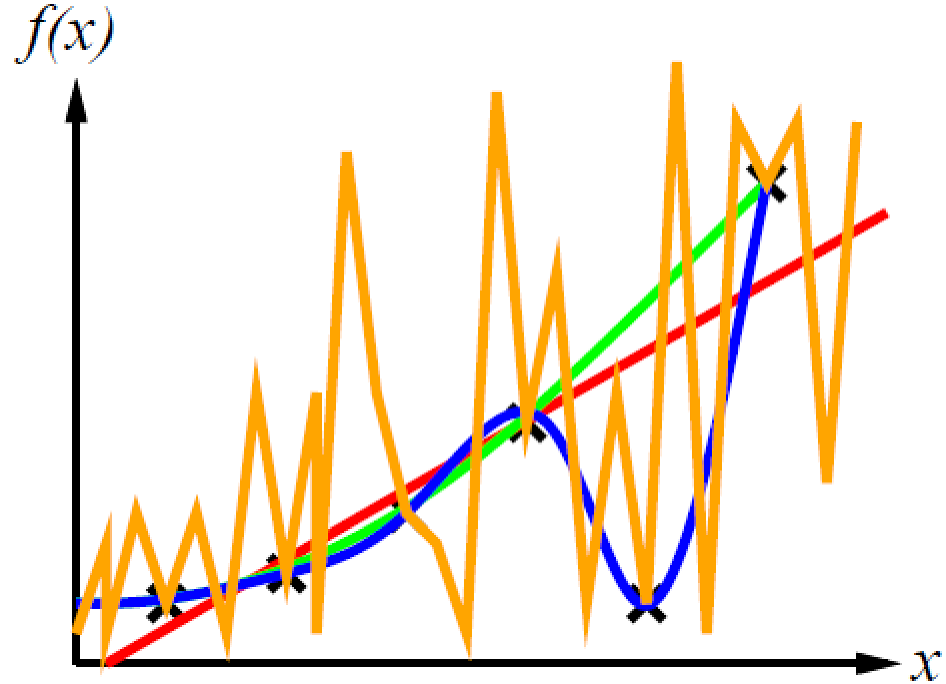
- What are the pitfalls of choosing the "best" classifier based on training error?

Supervised Learning Example



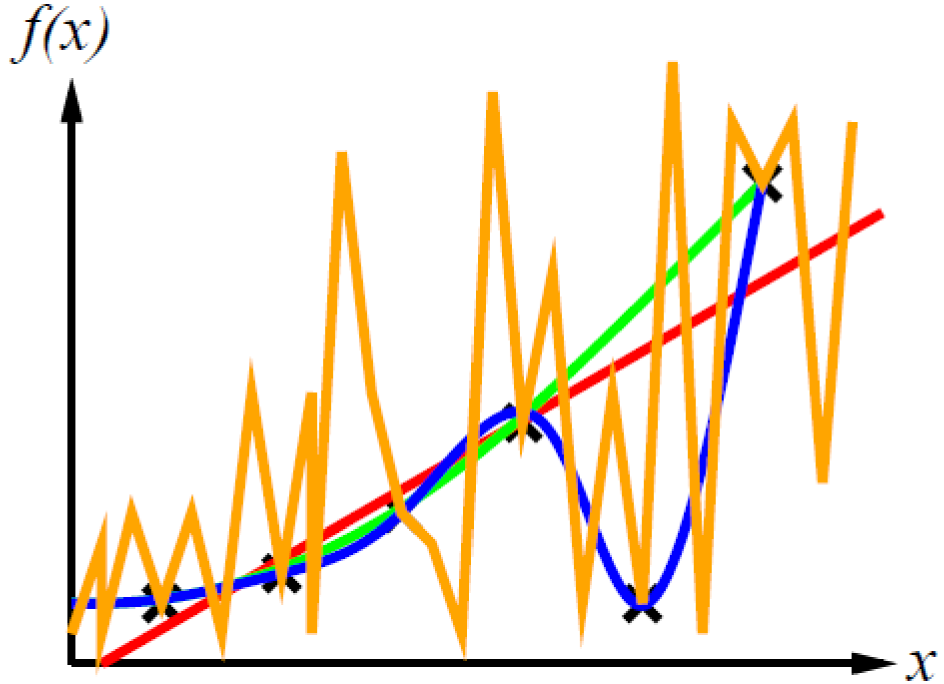
- What are the pitfalls of choosing the “best” classifier based on training error?
- The zig-zagging orange classifier comes out as “perfect”: its training error is zero.
- As a human, would you find more reasonable the orange classifier or the blue classifier (cubic polynomial)?
  - They both have zero training error.

Supervised Learning Example



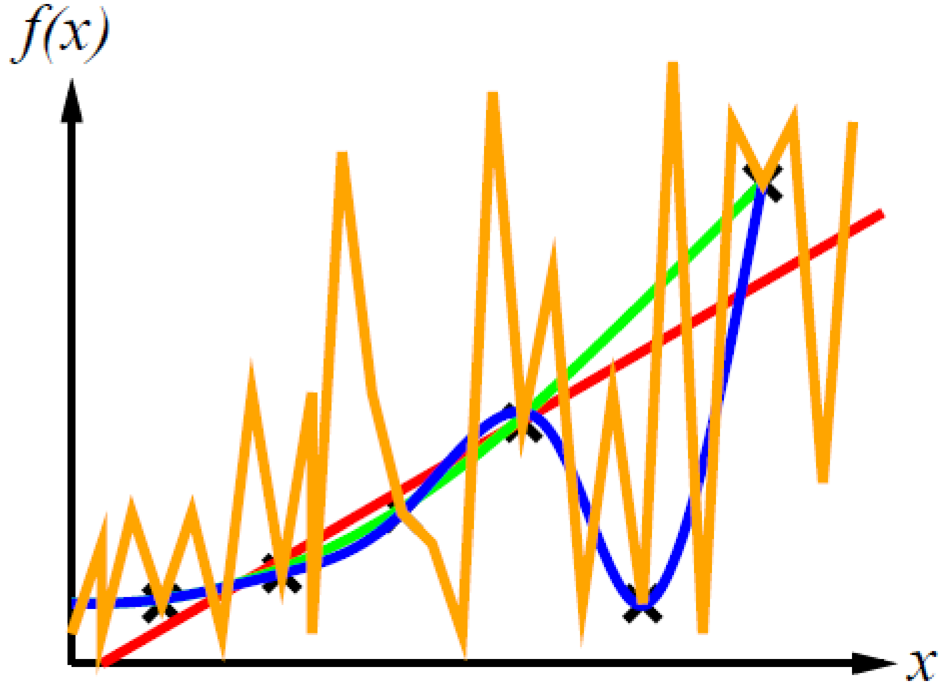
- What are the pitfalls of choosing the “best” classifier based on training error?
- The zig-zagging orange classifier comes out as “perfect”: its training error is zero.
- As a human, would you find more reasonable the orange classifier or the blue classifier (cubic polynomial)?
  - They both have zero training error.
  - However, the zig-zagging classifier looks pretty arbitrary.

Supervised Learning Example



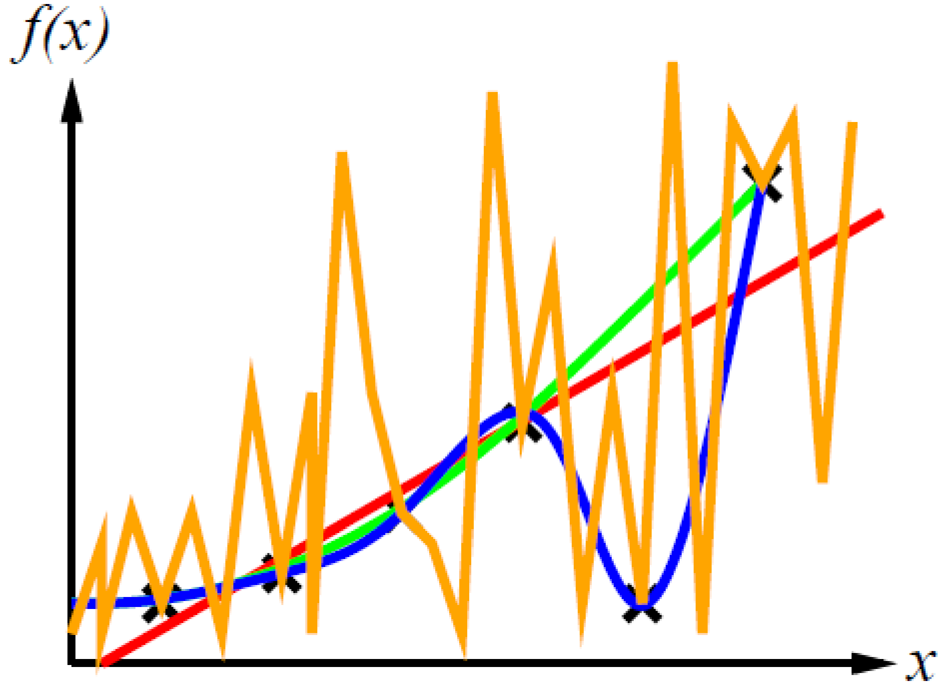
- Ockham's razor: given two equally good explanations, choose the more simple one.
  - This is an old philosophical principle (Ockham lived in the 14th century).
- Based on that, we prefer a cubic polynomial over a crazy zig-zagging classifier, because it is more simple, and they both have zero training error.

Supervised Learning Example



- However, real life is more complicated.
- What if none of the classifiers have zero training error?
- How do we weigh simplicity versus training error?

Supervised Learning Example



- However, real life is more complicated.
- What if none of the classifiers have zero training error?
- How do we weigh simplicity versus training error?
- There is no standard or straightforward solution to this.
- There exist many machine learning algorithms. Each corresponds to a different approach for resolving the trade-off between simplicity and training error.

The Road Ahead

- In the remainder of this course, we will mostly study supervised learning methods for pattern recognition.
- Some methods we will see, if we have time:
  - Decision trees.
  - Decision forests.
  - Bayesian classifiers.
  - Nearest neighbor classifiers.
  - Neural networks (in very little detail).
- Studying these methods should give you a good first experience with machine learning and pattern recognition.
- The current trend in AI is that machine learning and pattern recognition methods are becoming more and more dominant, with rapidly growing commercial applications and impact.