ECE 449/590 – OOP and Machine Learning
Lecture 14 Machine Learning Basics I

Professor Jia Wang
Department of Electrical and Computer Engineering
Illinois Institute of Technology

October 12, 2022

Outline

Learning Algorithms

Capacity and Generalization

Hyperparameters and Validation

Reading Assignment

- This lecture: Deep Learning 5
- Next lecture: Deep Learning 5

Outline

Learning Algorithms

Capacity and Generalization

Hyperparameters and Validation

What is machine learning?

- Machine learning: algorithms that are able to learn from data
  - Learning from experiences: performance at task T, as measured by P, improves with experience E.
- In comparison with traditional algorithm design, where
  - For task T, human learn from E and design/improve algorithms for better P.
  - But P will not improve for a specific algorithm when it is implemented.
  - E.g. to decide if a person smiles in an image.

The Task T

- Something that requires intelligence to complete.
  - Though we don't fully understand what is intelligence or have consensus on what count as artificial intelligence.
- Input: examples, each being a vector of features.
- Some common tasks
  - Classification and regression
  - Transcription, machine translation, and structured output.
  - Synthesis and sampling, anomaly detection, imputation of missing values
  - Density estimation, probability mass function estimation, denoising.

The Performance Measure P

- Quantitative measure of the abilities of machine learning algorithms.
- Choice of P depends on the task T.
  - Accuracy, or equivalently error rate and 0-1 loss, for tasks like classification that the output is either correct or not.
  - Otherwise, need to use continuous-valued scores like log-probability or mean squared error.
- Test set: it is preferable to apply P to data that the algorithm is not seen before.
- Many other factors to consider for complex tasks, e.g.
  - Partial credits
  - Worst-case vs average performance
  - Surrogates for hard-to-compute measures

The Experience E

- The ability to access examples.
- When the whole dataset of examples can be accessed,
  - Unsupervised learning: experience a dataset to learn useful properties like the probability distribution $p(\mathbf{x})$ for $\mathbf{x}$ being an example.
  - Supervised learning: each example is also associated with a label or target, and to learn the conditional distribution $p(\mathbf{y}|\mathbf{x})$ for $\mathbf{x}$ being an example and $\mathbf{y}$ being a label.
- Other variants
  - Semi-supervised learning and multi-instance learning.
  - Reinforcement learning: examples come from the environment, depending on how the learner learns to act there.

Dataset Representation

- As a design matrix $\boldsymbol{X}$.
  - Each row represents an example.
  - Each column corresponds to a feature.
- Labels, if presented, can be represented by a vector $\boldsymbol{y}$ where y_i is the label of example i.
- In practice, representations could be more complicated.
  - Examples may have different number of features.
  - Additional structures within features, e.g. image data usually contain x/y location and color channels.
  - Each label could be a sequence instead of single number.

Example: Linear Regression

- Supervised learning.
  - Each example $\boldsymbol{x} \in \mathbb{R}^n$ has a label $y \in \mathbb{R}$.
  - The model that predicts output from input: $\hat{y} = \boldsymbol{w}^\top \boldsymbol{x}$
  - $\boldsymbol{w} \in \mathbb{R}^n$ is a vector of parameters, or weights, to be learned.
- The task T: to predict y from $\boldsymbol{x}$ by outputting $\hat{y} = \boldsymbol{w}^\top \boldsymbol{x}$.
- The performance measure P
  - Test set of m examples: $\boldsymbol{X}^{(test)} \in \mathbb{R}^{m \times n}$
  - Labels: $\boldsymbol{y}^{(test)} \in \mathbb{R}^m$.
  - Prediction: $\hat{\boldsymbol{y}}^{(test)} = \boldsymbol{X}^{(test)} \boldsymbol{w}$
  - Measure the error using mean squared error (MSE)

$$\text{MSE}_{\text{test}} = \frac{1}{m} \sum_i (\hat{y}_i^{(test)} - y_i^{(test)})^2 = \frac{1}{m} ||\hat{\boldsymbol{y}}^{(test)} - \boldsymbol{y}^{(test)}||_2^2$$

Machine Learning Algorithm for Linear Regression

- One cannot learn from the test set.
- Training set of m' examples with labels: $(\boldsymbol{X}^{(train)}, \boldsymbol{y}^{(train)})$
- The best we can do is to find a $\boldsymbol{w}$ to minimize the error measure with respect to the training set.

$$\mathsf{MSE}_{\mathsf{train}} = \frac{1}{m'}||\hat{\boldsymbol{y}}^{(train)} - \boldsymbol{y}^{(train)}||_2^2$$

- Apply calculus to obtain $\boldsymbol{w}$:

$$\boldsymbol{w} = (\boldsymbol{X}^{(train)\top}\boldsymbol{X}^{(train)})^{-1}\boldsymbol{X}^{(train)\top}\boldsymbol{y}^{(train)}$$

A Linear Regression Problem with $n = 1$ and $m' = 10$

Linear Regression

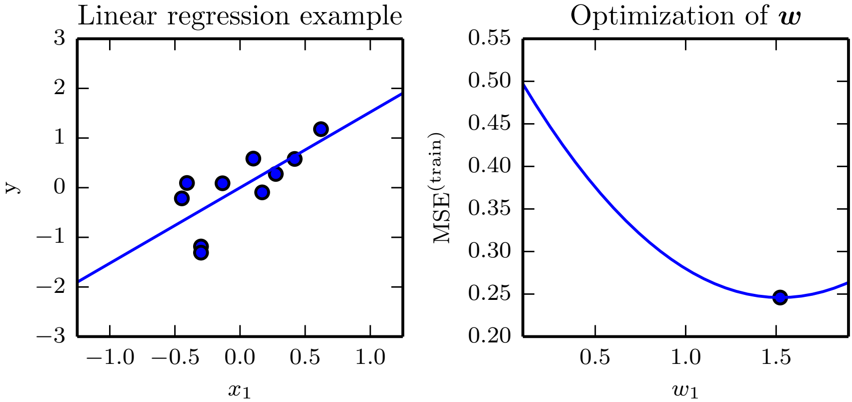


Figure 5.1

Discussions

- Usually a slightly more complicated prediction is used for linear regression: $\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} + b$
  - $b \in \mathbb{R}$ is the bias parameter to be learned.
- But what about the performance measure with respect to the test set?

Outline

Learning Algorithms

Capacity and Generalization

Hyperparameters and Validation

Generalization

- Training error, e.g. $\text{MSE}_{\text{train}}$
- Test error, or generalization error, e.g. MSE_{test}
- Generalization: the ability for a machine learning algorithm to perform well on previously unseen inputs.
  - Obviously $\text{MSE}_{\text{train}} \neq \text{MSE}_{\text{test}}$
  - Will a small $\text{MSE}_{\text{train}}$ imply a small MSE_{test}?
- Bayes error: error due to random noise assuming an ideal model.
  - Can training errors be smaller than Bayes errors?

Underfitting and Overfitting

- Since a machine learning algorithm has no direct control over test error, we may separate the need to minimize test error into two factors.
  1. Make the training error small.
  2. Make the gap between training and test error small.
- Underfitting: training error is too large.
- Overfitting: gap between the two is too large.
- Capacity: freedom of a model
  - E.g. dimension of the weights and bias
  - A model with lower capacity is more likely to underfit as there is not enough freedom to cover the whole training set.
  - A model with higher capacity is more likely to overfit as it tends to memorize the training set.

Underfitting and Overfitting in Polynomial Estimation

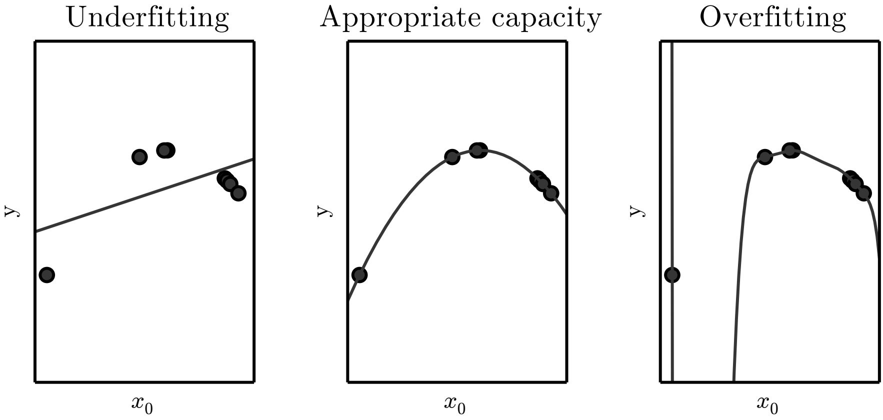


Figure 5.2

(Goodfellow 2016)

Generalization and Capacity

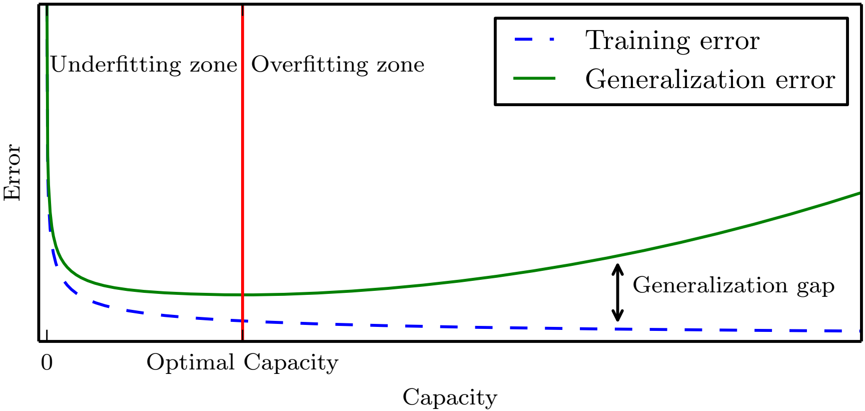
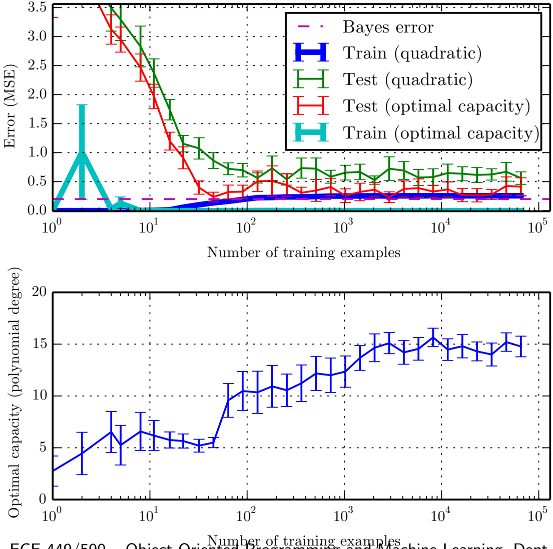


Figure 5.3

(Goodfellow 2016)

Training Set Size

Figure 5.4

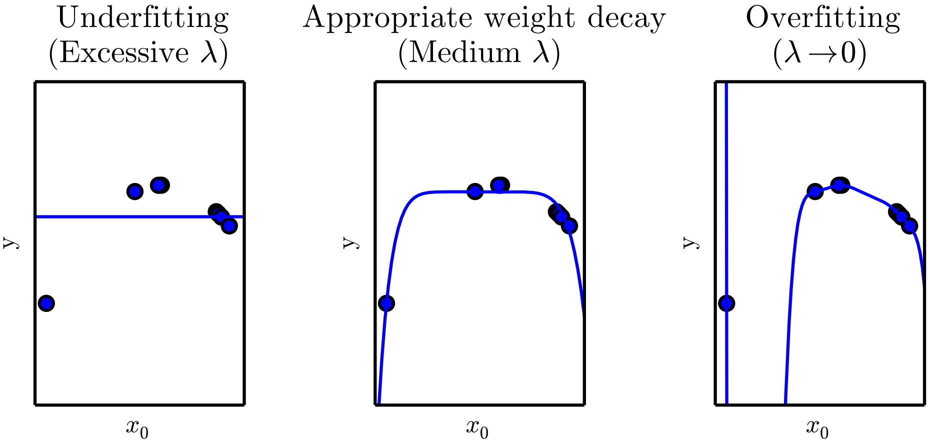


(Goodfellow 2016)

Regularization

- Regularization: impose certain preference on the model that reduces generalization error but not training error.
  - As if with reduced capacity.
- E.g. weight decay for linear regression.
  - Minimize $J(\boldsymbol{w}) = \mathsf{MSE}_{\mathsf{train}} + \lambda \boldsymbol{w}^\top \boldsymbol{w}$ instead of $\mathsf{MSE}_{\mathsf{train}}$
  - For $\lambda > 0$, prefer models with smaller $||\boldsymbol{w}||$, at the cost of higher $\mathsf{MSE}_{\mathsf{train}}$.
  - Larger training error, hopefully smaller test error.

Weight Decay



Underfitting (Excessive λ)

Appropriate weight decay (Medium λ)

Overfitting ($\lambda \to 0$)

Figure 5.5

(Goodfellow 2016)

Outline

Learning Algorithms

Capacity and Generalization

Hyperparameters and Validation

Hyperparameters

- Hyperparameters: settings that need to be decided before learning starts, e.g.
  - Degree of polynomial for polynomial regression
  - λ for weight decay
- How to decide good values for these hyperparameters?
  - Cannot use any example from test set.
  - Cannot use the whole training set as it will lead to overfitting.

Validation

- Validation set: a portion of training set that the learning algorithm will not experience.
- Validation: apply performance measure P to validation set to decide if the choice of hyperparamters is good.

Summary

- Supervised and unsupervised learning.
- Performance and error measure.
- Relation between capacity, underfitting, and overfitting.
- Tune hyperparameters with validation set.