https://ttpoll.com/p/389226

COSC 325: Introduction to Machine Learning

Dr. Hector Santos-Villalobos

Dr. Santos

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Lecture 05 - Learning Theory and Gradient Descent

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Class Announcements

Homework:

We will release the key by the end of the week.

Exams:

Exam #1: (1) Online, (2) Time-bounded 1 hr, (3) From 11 am to 1 pm.

Lectures:

The October 10^{th} lecture will be online.

Course Project:

Team assignments by the end of the week. Check Canvas->People.

What: UTK Machine Learning Club

Where: MK 525

When: Tuesday at 5:00
(including today)

Who: Any experience level

Everyone is welcome to the first meeting of ML club today. Whether you are a beginner looking to learn from our intro to ML lesson series, experienced practitioner who wants to learn from and discuss with other enthusiasts in our reading groups, or you just want to hear from our industry guest speakers and seminars, utkML can help you scratch your machine learning itch!

Today’s Topics

Learning Theory

Gradient Descent

Last Lecture

- Pandas
  - Excellent tool for data preprocessing
- Scikit-learn
  - Playground for everything ML

Pop Quiz

1 | MULTIPLE CHOICE

How long did it take you to complete homework #1?

A. Less than one hour.

B. 1 - 2 hrs

C. 2 - 3 hrs

D. More than 3 hours

ML Life Cycle

Use Case / Application
- Cancer detection
- Clustering
- Object segmentation
- Control of pressure valve

Machine Learning Category
- Supervised
- Self-supervised
- Semi-supervised
- Reinforcement

Data
- Data acquisition
- Training, validation, test data split
- Data Wrangling
  - Exploratory Data Analysis (EDA)
  - Data Scaling
  - Data cleaning
  - Feature extraction and selection

Machine Learning Technique
- Specific technique
  - Linear Regression
  - Multi-layer Perceptrons (MLP)
  - KNNs
- Objective Functions (ML Training)
- Hyperparameter tuning

Evaluation
- Bias/Variance Analysis
- Cross-Validation
- Performance Metric (Application)
- Explainability
- Fairness, Transparency, and Privacy

Deployment
- Stress test
- Key Performance Indicators (KPIs)
- Model Monitoring
  - Data drift
- Model Refresh

ML Life Cycle

Use Case / Application
- Cancer detection
- Clustering
- Object segmentation
- Control of pressure valve

Machine Learning Category
- ***Supervised***
- Self-supervised
- Semi-supervised
- Reinforcement

Data
- Data acquisition
- Training, validation, test data split
- Data Wrangling
  - Exploratory Data Analysis (EDA)
  - Data Scaling
  - Data cleaning
  - Feature extraction and selection

COSC 426

Machine Learning Technique
- Specific technique
  - Linear Regression
  - Multi-layer Perceptrons (MLP)
  - KNNs
- Objective Functions (ML Training)
- Hyperparameter tuning

Evaluation
- Bias/Variance Analysis
- Cross-Validation
- Performance Metric (Application)
- Explainability
- Fairness, Transparency, and Privacy

Deployment
- Stress test
- Key Performance Indicators (KPIs)
- Model Monitoring
  - Data drift
- Model Refresh

Examples of ML Techniques

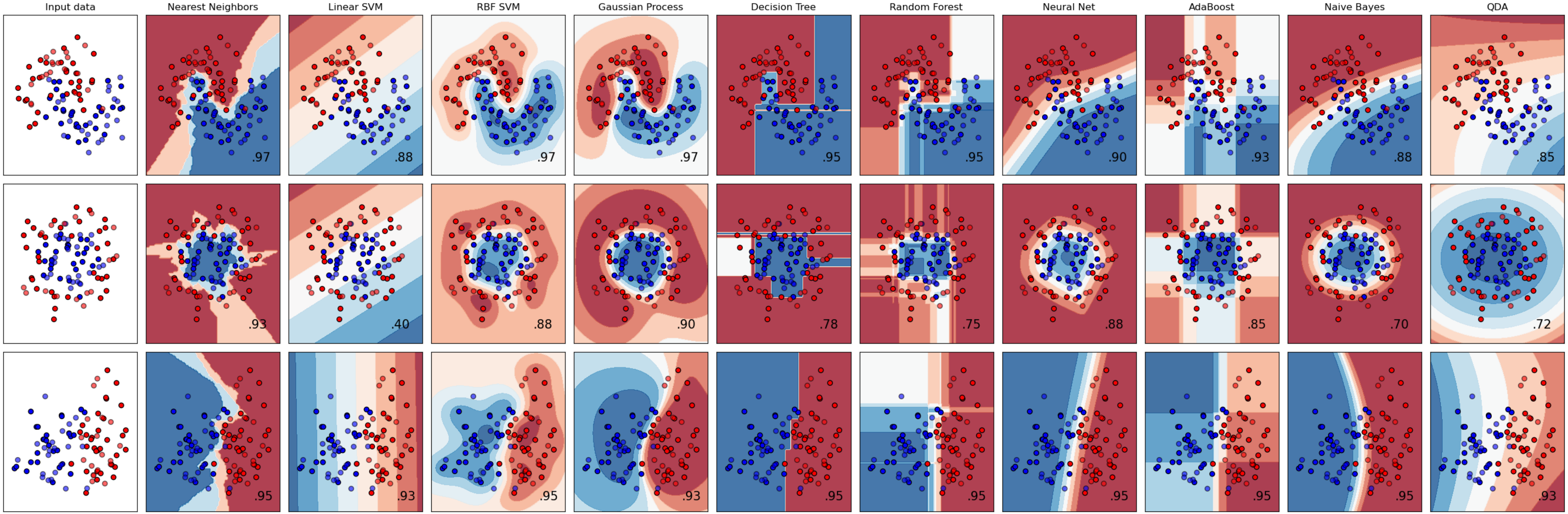
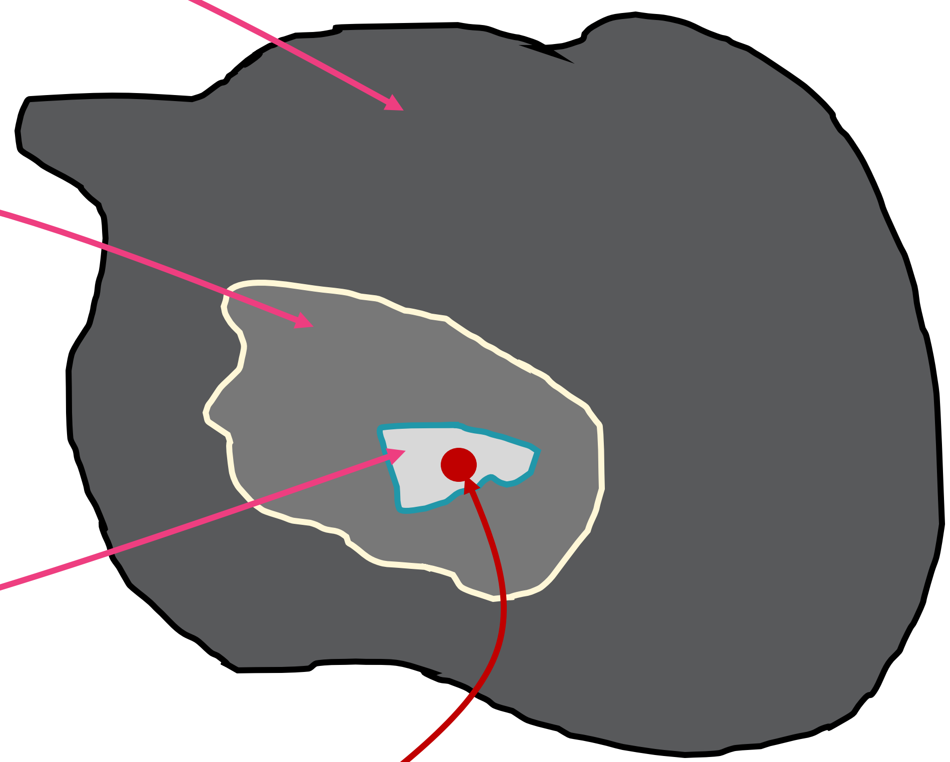


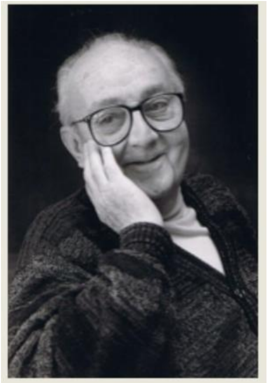
Image source: https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

RBF – Radial Basis Function
QDA – Quadratic Discriminant Analysis

Hypothesis Space



A model cannot make a better hypothesis than one provided by the sample distribution and within the limits of the learning category and technique.



"All models are wrong, but some are useful."
– Prof. George Box

What does $f(x, y)$ tell us?

- $f(x, y)$ is the true hypothesis probability distribution for data in f
- If features/target pair (x, y)
  - Belongs to f, then, f will return a high probability ~ 1.
  - Does not belong to f, then, f will return a low probability ~ 0.
- Example: Assume f is the probability distribution of images of an object x and the corresponding label y.

What does $f(x, y)$ tell us?

- If features/target pair (x, y)
  - Belongs to f, then, f will return a high probability ~ 1.
  - Does not belong to f, then, f will return a low probability ~ 0.
- Example: Assume f is the probability distribution of images of an object x and the corresponding label y.

y	"Cat"
x	
$f(x, y)$	0.98

THE UNIVERSITY OF TENNESSEE KNOXVILLE

What does $f(x, y)$ tell us?

- If features/target pair (x, y)
  - Belongs to f, then, f will return a high probability ~ 1.
  - Does not belong to f, then, f will return a low probability ~ 0.
- Example: Assume f is the probability distribution of images of an object x and the corresponding label y.

y	"Cat"	"Car"
x		
$f(x, y)$	0.98	0.95

https://ttpoll.com/p/389226

What does $f(x,y)$ tell us?

- If features/target pair (x,y)
  - Belongs to f, then, f will return a high probability ~ 1.
  - Does not belong to f, then, f will return a low probability ~ 0.
- Example: Assume f is the probability distribution of images of an object x and the corresponding label y.

y	"Cat"	"Car"	"Mountain Lion"
x			
$f(x,y)$	0.98	0.95	0.32

Pop Quiz

1 | MULTIPLE CHOICE POINTS: 1 | Edit

Although classical machine learning techniques cannot accurately model the true data distribution f, advanced deep learning techniques can.

A. True

B. False

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Pop Quiz

1 | MULTIPLE CHOICE

POINTS: 1 | Edit

Although classical machine learning techniques cannot accurately model the true data distribution f, advanced deep learning techniques can.

A. True

B. False

A model cannot make a better hypothesis than one provided by the sample distribution and within the limits of the learning category and technique.

This applies to DL also.

Important Note

- We make ***NO*** assumptions about what the distribution f looks like or what it is
  - If we did know, it would make our learning problem easier!
- We can only get a random sample from f
  - This is our training data!

Slide credit: Dr. Schuman

Bayes Optimal Classifier

When we know f.

Data Generating Distributions

- The underlying assumption is that learning problems are characterized by some unknown probability distribution f over input/output pairs (x, y)
- Suppose we know what f is
  - If we have a density function that takes x and y and produces a probability of that pair in f
- If we have that, classification becomes easy (Bayesian Optimal Classifier):

$$\hat{y} = h^{BO}(x) = \arg\max_{y \in C} f(x, y)$$

C is the set of possible targets.

Theorem #1: Bayes Optimal Classifier

- Bayesian optimal classifier:

$$h^{BO}(x) = \arg\max_{y \in C} f(x, y)$$

- Theorem 1: The Bayes Optimal Classifier $h^{(BO)}$ achieves minimal zero/one error of any deterministic classifier.
  - Note: This assumes comparison against deterministic classifiers $(\hat{y}^{(i)} = h(x^{(i)}))$

$$0\text{–}1 \text{ loss} = \text{Zero/One Error} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left(\hat{y}^{(i)} \neq y^{(i)}\right)$$

Slide credit: Dr. Schuman

Proof of Theorem #1

- Suppose you have a classifier g that claims to be better than $h^{(BO)}$
- There must be x on which $g(x) \neq h^{(BO)}(x)$.
- Probability that $h^{(BO)}$ makes an error on this particular x is:

$$1 - f\left(x, h^{(BO)}(x)\right)$$

- Similarly, the probability that g makes an error on this particular x is:

$$1 - f\left(x, g(x)\right)$$

Slide credit: Dr. Schuman

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Proof of Theorem #1 (Continued)

- However, $h^{(BO)}$ was chosen so that it maximizes $f\left(x, h^{(BO)}(x)\right)$, thus:

$$f\left(x, h^{(BO)}(x)\right) > f\big(x, g(x)\big) \Rightarrow 1 - f\left(x, h^{(BO)}(x)\right) < 1 - f\big(x, g(x)\big)$$

- So, the probability that $h^{(BO)}$ is wrong on this x is smaller than that of g on this x.
  - This applies to any x for which $g(x) \neq h^{(BO)}$
- Thus, $h^{(BO)}$ achieves smaller zero/one error than any g.
- QED (Quod Erat Demonstrandum)

Notebook Time

Pop Quiz

1 | MULTIPLE CHOICE POINTS: 1

What technique offers the optimal error rate when the data distribution f is unknown?

A. Bayes Optimal Classification

B. Deep Learning

C. K-Nearest Neighbors

D. Logistic Classification

E. All of the above

F. None of the above

Pop Quiz

1 | MULTIPLE CHOICE POINTS: 1

What technique offers the optimal error rate when the data distribution f is unknown?

A. Bayes Optimal Classification

B. Deep Learning

C. K-Nearest Neighbors

D. Logistic Classification

E. All of the above

F. None of the above

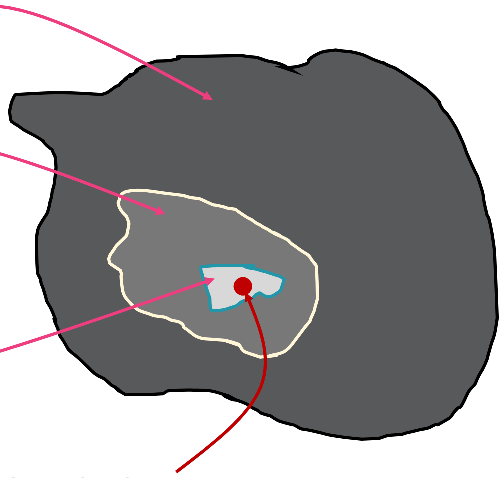
Bayes Optimal Error Rate

- The best error rate you can ever hope to achieve on a particular classification problem.
- Building the optimal classifier would be trivial if someone gave you the data distribution f.
- We don't have that, so we must figure out how to build a classifier h with a training set sampled from f.

Important Reminder

We can ***NEVER*** expect a machine learning algorithm to generalize beyond the data distribution, the learning category, and the learning technique.

Hypothesis Space



Review

- Hypothesis f:
- Model h_θ:
- θ:
- Learning algorithm:
- Objective function J_θ:

Review

- **Hypothesis f:** A hypothesis is a certain function that we believe (or hope) is similar to the true function, the target function we want to model.
- **Model h_θ:** In the machine learning field, the terms hypothesis and model are often used interchangeably. In other sciences, they can have different meanings.
- **θ:** The learned parameters for model h_θ.
- **Learning algorithm:** Again, our goal is to find or approximate the target function, and the learning algorithm is a set of instructions that tries to model the target function using our training dataset. A learning algorithm comes with a hypothesis space, the set of possible hypotheses it explores to model the unknown target function by formulating the final hypothesis. It is also called learning technique.
- **Objective function J_θ**: Often synonymously with loss $\mathcal{L}_\theta$ or cost function; sometimes called error function, empirical risk, or training error. In some contexts, the loss is for a single data point, whereas the objective function refers to the expected error/loss over the entire dataset.

The relationship between the expected value and the cost $J(\theta)$

Loss and Cost Functions

- Loss $\mathcal{L}_\theta\left(y^{(i)}, \hat{y}^{(i)}\right)$ is the error between the ground truth (i.e., expected response) $y^{(i)}$ and the model prediction $\hat{y}^{(i)}$.

Per sample $x^{(i)}$

- Cost $J(\theta)$ is a measure of overall model error for parameters θ.

Expected Performance

We want both to be SMALL

Expected Value

- Expectation means “average”
- If you draw a bunch of (x, y) pairs independently at random from f, what would your average loss be?

$$E_{x,y\sim f}\left[f(x,y)\mathcal{L}\big(y,h(x)\big)\right]$$

Expected Value

- Weighted average loss over all (x, y) pairs in f, weighted by their probability $f(x, y)$. If $f(x, y)$ is a finite discrete distribution, e.g., defined by a finite data set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ that puts equal weight on each example

$$E_{x,y\sim f}\left[f(x,y)\mathcal{L}\big(y,h(x)\big)\right] = \sum_{(x,y)\in f}\left[f(x,y)\mathcal{L}\big(y,h(x)\big)\right]$$

$$= \sum_{i=1}^{n}\left[f\left(x^{(i)},y^{(i)}\right)\mathcal{L}\left(y^{(i)},h\left(x^{(i)}\right)\right)\right] = \sum_{i=1}^{n}\left[\frac{1}{n}\mathcal{L}\left(y^{(i)},h\left(x^{(i)}\right)\right)\right] = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\left(y^{(i)},h\left(x^{(i)}\right)\right)$$

Slide credit: Dr. Schuman

Loss and Cost Functions

- Loss $\mathcal{L}_\theta\left(y^{(i)}, \hat{y}^{(i)}\right)$ is the error between the ground truth (i.e., expected response) $y^{(i)}$ and the model prediction $\hat{y}^{(i)}$.
- Cost $J(\theta)$ is a measure of overall model error for parameters θ.

Per sample $x^{(i)}$

Expected Performance

We want both to be SMALL

The average loss $\mathrm{J}(\theta) = \frac{1}{\mathrm{n}}\sum_{i=1}^{n} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$.

Connecting the data (x, y) to the loss $\mathcal{L}_{\boldsymbol{\theta}}$.

How do computers learn?

Learning Problem Definition

- Learning problem defined by:
  - The loss $\mathcal{L}_\theta(y, h_\theta(x))$ function, which captures our notion of what is important to learn
  - The data generating distribution f, which defines the data we expect to see
- Based on the training data, we ***induce*** a function $h_\theta(x)$ that maps new inputs x to predictions $\hat{y}$
- h should do well (based on the loss function) on future examples that are ALSO drawn from f
- We care about f, but we don't know f.

Slide credit: Dr. Schuman

ML Induction

- Formal definition of ***induction machine learning***:

Given (i) a loss function $\mathcal{L}_\theta$ and (ii) a sample from some unknown distribution f, you must compute a function h that has low expected error over f w.r.t. $\mathcal{L}_\theta$.

Slide credit: Dr. Schuman

Regression Objective Function Candidates

- Sum of Squared Residuals (Very similar to MSE)

$$\text{SSR} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

Convex objective function (i.e., a local minimum is a global minimum)

Makes small residuals even smaller

Makes large residuals explode in magnitude.

It is by far the most popular objective function for regression problems.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Regression Objective Function Candidates

- Sum of Squared Residuals (Very similar to MSE)

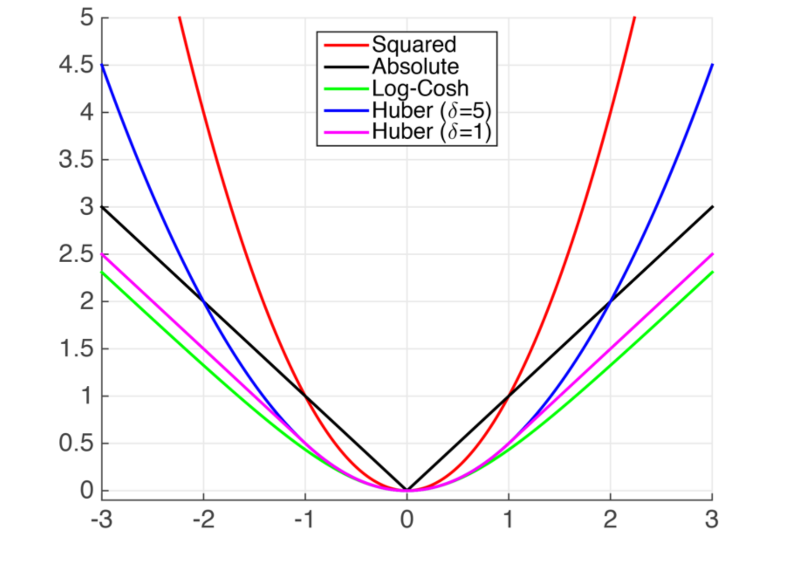
$$\text{SSR} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

Non-differentiable at zero (i.e., $y^{(i)} == \hat{y}^{(i)}$)

- Mean Absolute Error

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y^{(i)} - \hat{y}^{(i)} \right|$$

Less sensitive to outliers



Regression Objective Function Candidates

- Sum of Squared Residuals (Very similar to MSE)

$$\text{SSR} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

- Mean Absolute Error

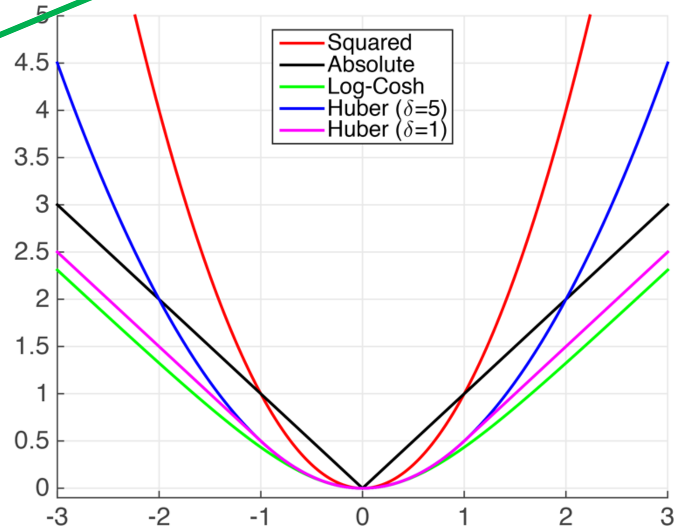
$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y^{(i)} - \hat{y}^{(i)}|$$

- Huber Loss

$$\text{Huber Loss} = \sum_{i=1}^{n} L_\delta(y^{(i)} - \hat{y}^{(i)})$$

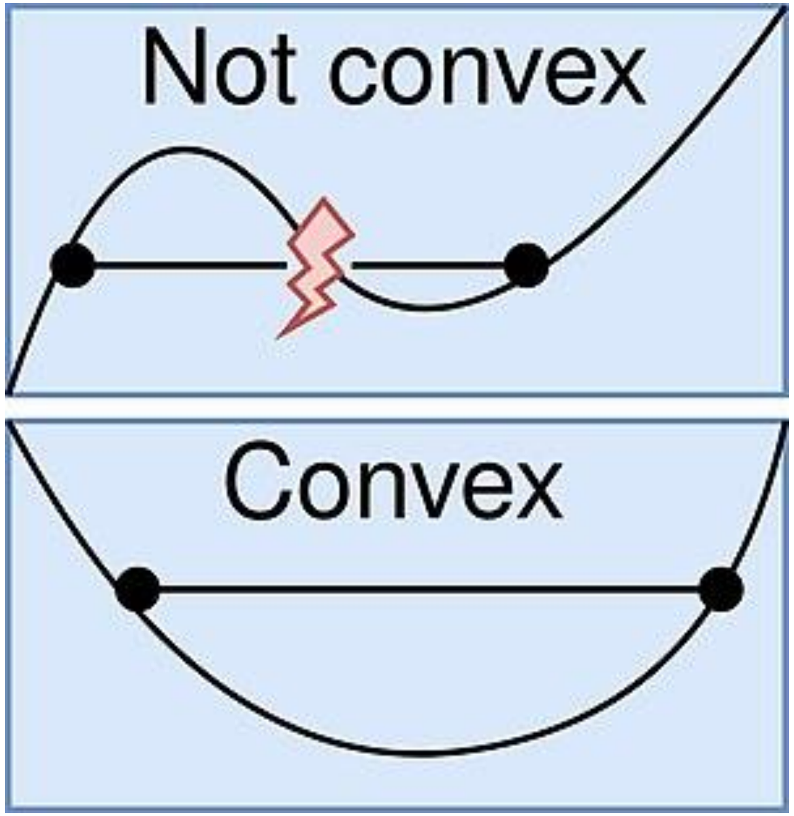
$$\text{where } L_\delta(r) = \begin{cases} \frac{1}{2} r^2 & \text{if } |r| \le \delta \\ \delta(|r| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

Combines the strengths of SSR and MAE (i.e., quadratic for small errors and linear for large errors)

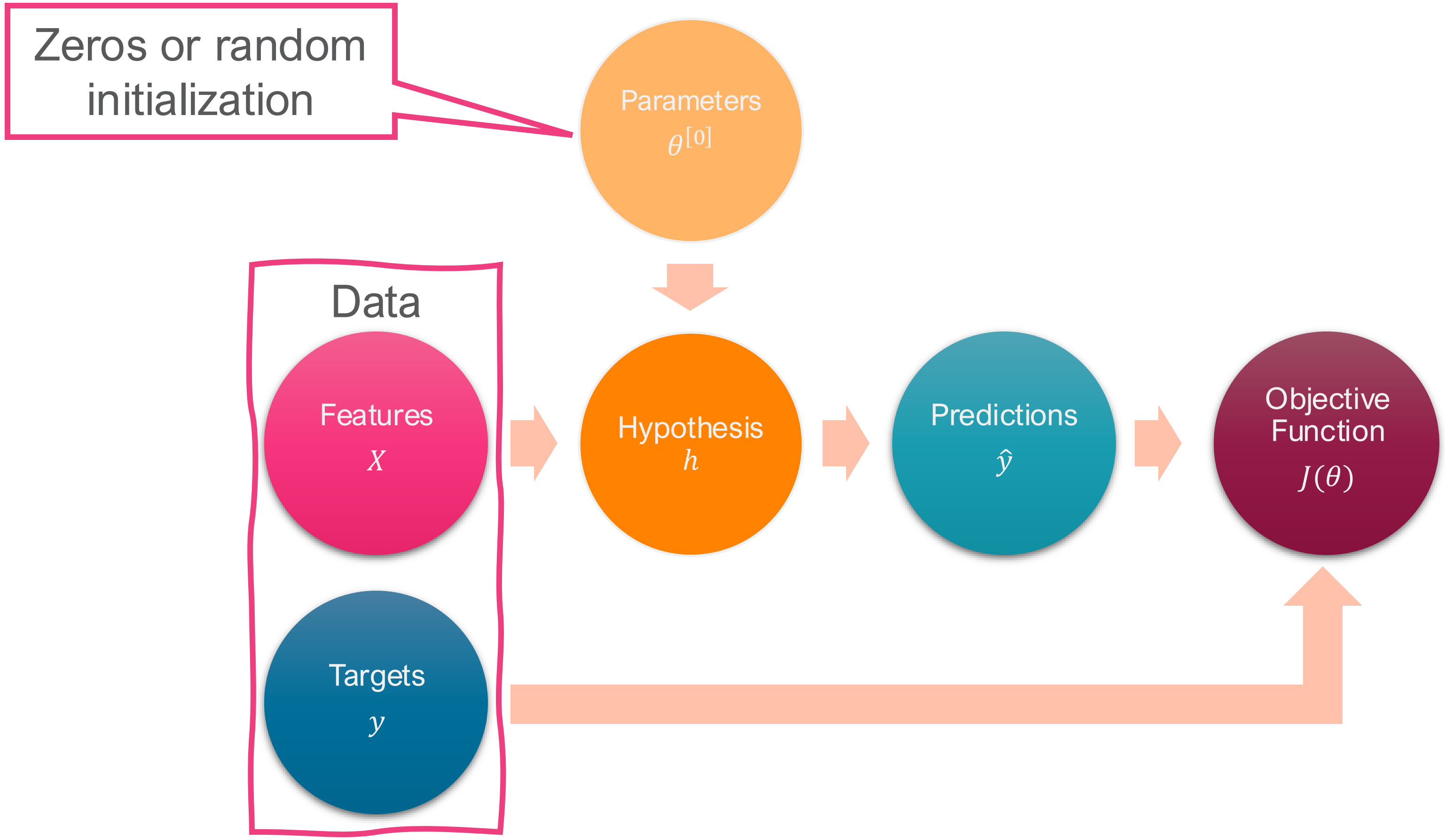


Preferred Objective Functions Characteristics

- Adequate sensitivity to outliers
- Computationally efficient
- Differentiable everywhere
- Interpretable
- Convex
- Aligned with the use case

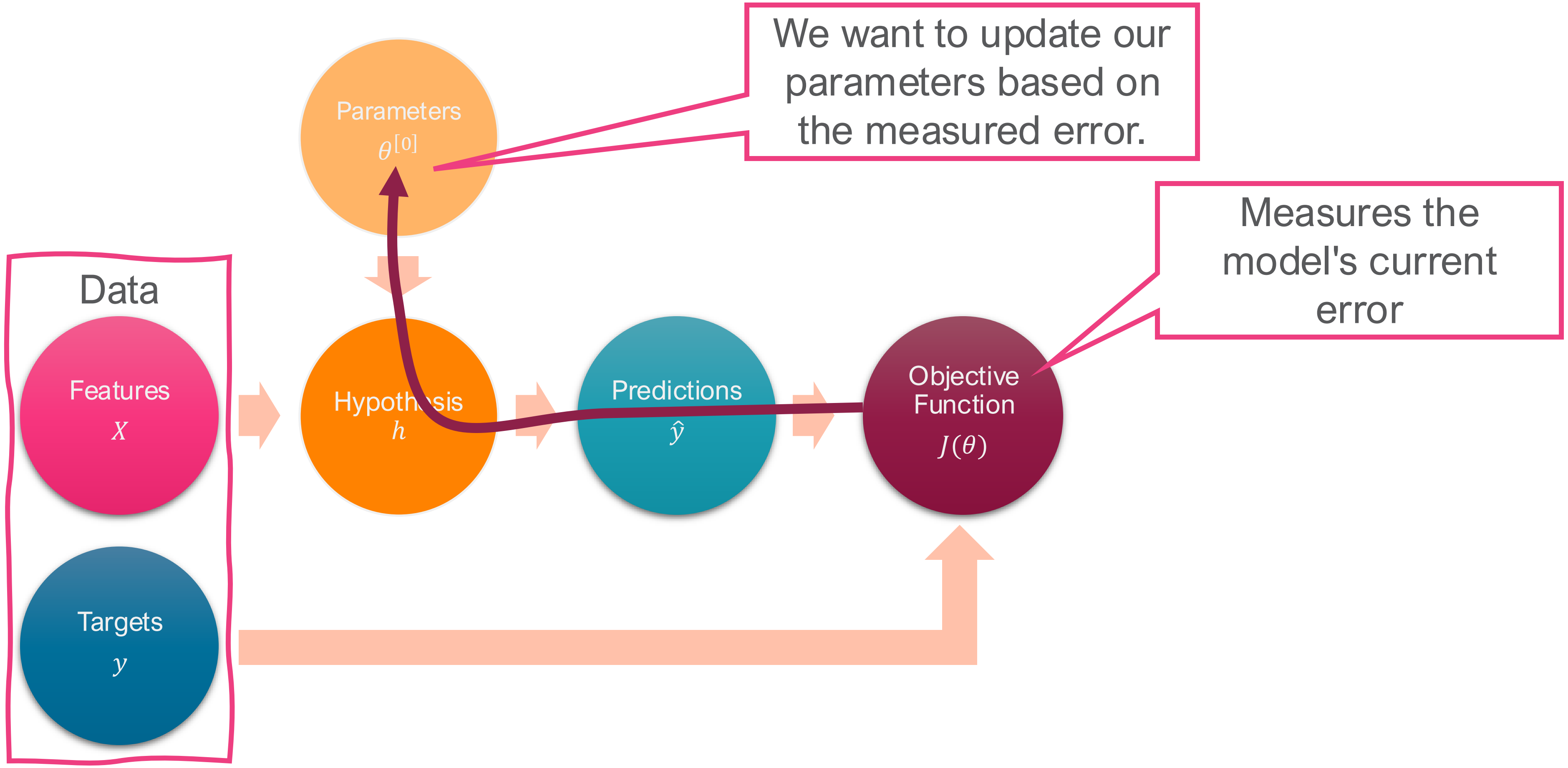


How do computers learn?



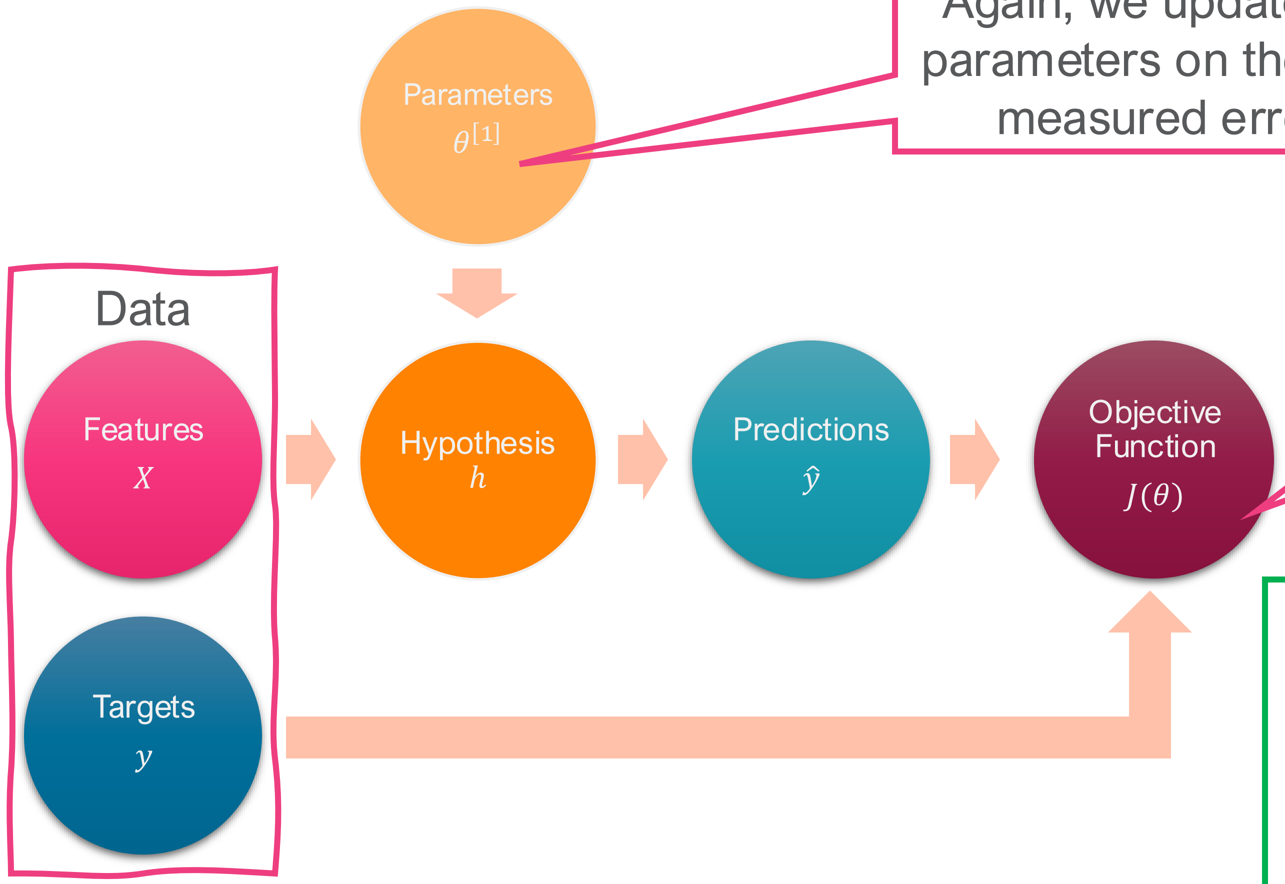
THE UNIVERSITY OF TENNESSEE KNOXVILLE

How do computers learn?



THE UNIVERSITY OF TENNESSEE KNOXVILLE

How do computers learn?



Again, we update the parameters on the new measured error.

We can test our updated parameters for a new measurement of error.

We repeat (i.e., iterate) this process until the error is small enough or we reach a maximum number of iterations.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

How does this magic happen?

How can we learn θ?

Gradient Descent

Gradient Descent

- First-order optimization (find minimum or maximum) technique
  - Only the first derivative is needed.
- Moves in the direction of steepest descent/accent
- It is the most popular method to minimize the error in the cost $J(\theta)$
- Types of GD
  - Batch: all samples are used for each update (i.e., iteration)
  - Stochastic (SGD): one sample per parameter update
  - Mini-Batch: a subset of the batch is used per iteration
    - Typical values: **32, 64, 128**, 256

Gradient Descent Algorithm

$X \coloneqq \text{data features}$

$y \coloneqq \text{data targets}$

$\theta = \theta_0$

Repeat:

$$\hat{y} = h_\theta(X)$$

$$cost = J_\theta(y, \hat{y})$$

$$d\theta = \frac{\partial J_\theta\,(y,\hat{y})}{\partial \theta}$$

$$\theta \coloneqq \theta - \alpha(d\theta)$$

Until a fixed number of iterations or $d\theta$ very small.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Review

- A model cannot make a better hypothesis than one provided by the sample distribution and within the limits of the learning category and technique.
- Bayes Optimal Classifier is the best solution when the data distribution f is known.
- Gradient descent
  - An iterative process to minimize model error
  - Simplicity is King
  - Needs the first derivative of the cost w.r.t the parameters
  - A derivative tells us the influence of a parameter on the cost

Next Lecture

- We will apply these concepts to
  - Linear regression
  - Polynomial regression
  - Logistic regression and classification

What: UTK Machine Learning Club

Where: MK 525

When: Tuesday at 5:00
(including today)

Who: Any experience level

Everyone is welcome to the first meeting of ML club today. Whether you are a beginner looking to learn from our intro to ML lesson series, experienced practitioner who wants to learn from and discuss with other enthusiasts in our reading groups, or you just want to hear from our industry guest speakers and seminars, utkML can help you scratch your machine learning itch!