Linear Regression

Dr. Mahmoud Nabil

Dr. Mahmoud Nabil
mnmahmoud@ncat.edu

North Carolina A & T State University

September 12, 2022

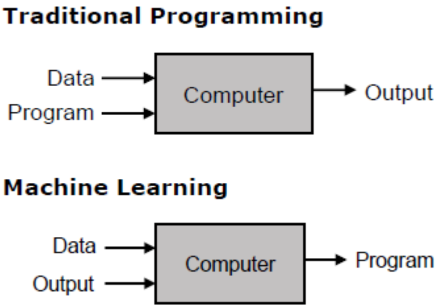
Talk Overview

1. ML inroduction
2. ML Terminologies
3. ML Applications Examples
4. Linear Regression

Outline

So What is Machine Learning?

- Automating automation
- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!



Machine Learning Problem Types

- **Based on Type of Data**
    - Supervised, Unsupervised, Semi supervised, Reinforcement Learning
- **Based on Type of Output**
    - Regression, Classification
- **Based on Type of Model**
    - Generative, Discriminative

We will focus on: Supervised → {Regression, Classification} → Discriminative models

Types of Learning based on Type of Data

- Supervised learning
    - Training data includes desired outputs.
    - Trying to learn a relation between input data and the output
- Unsupervised learning
    - Training data does not include desired outputs.
    - Trying to “understand” the data.
- Semi supervised learning
    - Training data includes a few desired outputs.
- Reinforcement learning
    - Rewards from sequence of actions.

Types of Learning based on Type of Output

Regression

A regression model predicts continuous values.

For example:

- What is the value of a house in California?
- What is the probability that a user will click on this ad?

Classification

A classification model predicts discrete values.

For example:

- Is a given email message spam or not spam?
- Is this an image of a dog, a cat, or a hamster?

Types of Learning based on Type of Model

Generative Model

A Generative nodel explicitly learns the actual distribution of each class.

Discriminative Model

A Discriminative model learns the decision boundary between the classes.

Generative Models

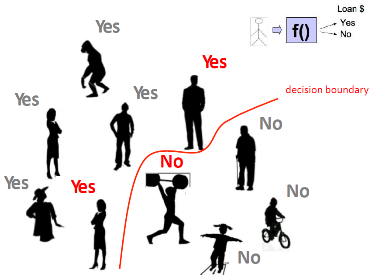
- Naïve Bayes
- Hidden Markov Models
- Bayesian networks
- Markov random fields

Discriminative Models

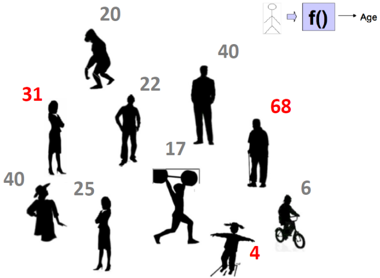
- Logistic regression
- SVMs
- Traditional neural networks
- Nearest neighbor
- Conditional Random Fields (CRF)

Regression vs Classification

Classification (supervised learning)



Regression (supervised learning)



Outline

ML Basic Terminologies

Many terminologies associated with ML. Will cover them in the following slides.

- Labels
- Features
- Examples
- Models

Label

- A label is the thing that we are predicting in classification or regression task. **Example:** Male, Female
- The label could also be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.
- Usually denoted with the variable y.

Features (1/2)

- A feature is an input variable (a.k.a attribute).
- A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features.
- Usually denoted as:

$$x_1, x_2, \ldots, x_N$$

In the spam detector example, the features could include the following:

- words in the email text
- sender's address
- time of day the email was sent
- ...

Features (2/2)

Generally three types of attributes:

- **Categorical:** red, blue, brown, yellow
- **Ordinal:** poor, satisfactory, good, excellent
- **Numeric:** 3.14, 6E23, 0,

Features (2/2)

Generally three types of attributes:

- **Categorical:** red, blue, brown, yellow
- **Ordinal:** poor, satisfactory, good, excellent
- **Numeric:** 3.14, 6E23, 0,

Categorical

- No natural ordering to categories
- Categories usually encoded as numbers

Ordinal

- There is a natural ordering to categories
- Encoded as numbers to preserve ordering

Numeric

- Integers or real numbers
- meaningful to add, mul tiply, compute

Notethat

The process of generating this features for our machine learning problem is called feature engineering.

Data samples (Examples)

Data sample / Example is a particular instance of data, **x**. (Note That. **x** is a vector of features)

We break examples into two categories:

- Labeled examples: (Used for prediction)
- Unlabeled examples: (Used for inference/testing)

Example

housingMedianAge (feature)	totalRooms (feature)	totalBedrooms (feature)	medianHouseValue (label)
15	5612	1283	66900
19	7650	1901	80100
17	720	174	85700
14	1501	337	73400
20	1454	326	65500

Model

A **model** defines a relationship between features and label.

Two phases of a model's life:

- **Training** means creating or learning the model. You show the model labeled examples and enable the model to gradually learn the relationships between features and label.
- **Testing/Inference** means applying the trained model to unlabeled examples. You use the trained model to make useful predictions (y').

Outline

1. ML inroduction
2. ML Terminologies
3. ML Applications Examples
4. Linear Regression

ML Application 1: Credit Approval

- **Numeric Features:**
  - loan amount (e. g. $1000)
  - Income (e. g. $65000)
- **Ordinal Features:**
  - savings: {none, <100, 100..500, 500..1000, >1000}
  - employed: {unemployed, <1yr, 1..4yrs, 4..7yrs, >7yrs}
- **Categorical Features:**
  - purpose: {car, appliance, repairs, education, business}
  - personal: {single, married, divorced, separated}
- **Labels (Categorical):**
  - Approve credit application
  - Disapprove credit application

ML Application 1: Credit Approval

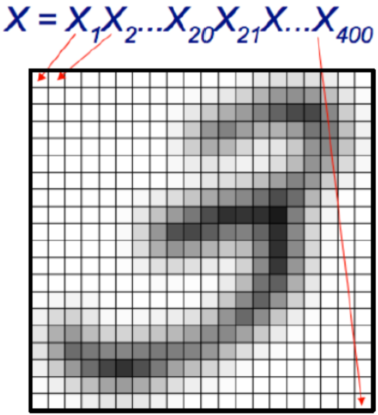
- **Numeric Features:**
  - loan amount (e. g. $1000)
  - Income (e. g. $65000)
- **Ordinal Features:**
  - savings: {none, <100, 100..500, 500..1000, >1000}
  - employed: {unemployed, <1yr, 1..4yrs, 4..7yrs, >7yrs}
- **Categorical Features:**
  - purpose: {car, appliance, repairs, education, business}
  - personal: {single, married, divorced, separated}
- **Labels (Categorical):**
  - Approve credit application
  - Disapprove credit application

Easy feature engineering process.

ML Application 2: Handwritten Digits Recognation

Represent each pixel as a separate attribute either Categorical **OR** Ordinal:

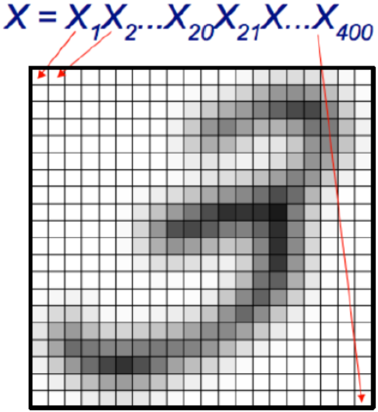
- **Categorical Features:**
  - (white) or (black) based on a threshold
- **Ordinal Features:**
  - Degree of "blackness" of a pixel
- **Labels (Categorical):** {0,1,2,3,4,5,6,7,8,9}

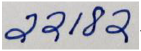


ML Application 2: Handwritten Digits Recognation

Represent each pixel as a separate attribute either Categorical **OR** Ordinal:

- **Categorical Features:**
    - (white) or (black) based on a threshold
- **Ordinal Features:**
    - Degree of "blackness" of a pixel
- **Labels (Categorical):** {0,1,2,3,4,5,6,7,8,9}



What if we are dealing with paper like this 

?

Isolate each digit, rescale, de-slant, ..

Hard feature engineering process.

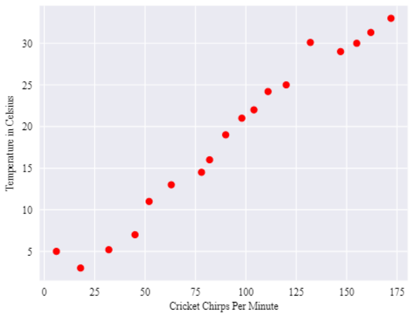
Outline

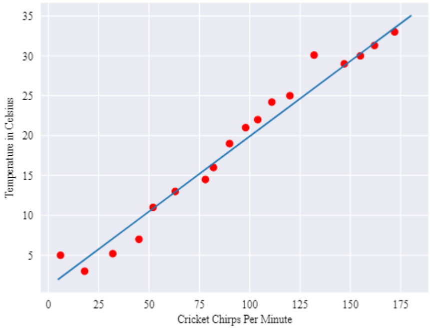
Linear Regression

Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data

Example: Scientists found that crickets (an insect species) chirp more frequently on hotter days than on cooler days.





Linear Regression

A linear relationship

- True, the line doesn't pass through every dot.
- However, the line does clearly show the relationship between chirps and temperature.

$$y = mx + b$$

where:

- y: is the temperature in Celsius—the value we're trying to predict.
- m: is the slope of the line.
- x: is the number of chirps per minute—the value of our input feature.
- b: is the y-intercept.

Linear Regression

In machine learning, we'll write the equation for a model slightly differently:

$$y' = w_1 x_1 + w_0$$

where:

- y': is the predicted label (a desired output).
- w_1: is the weight of feature 1. Weight is the same concept as the "slope".
- x_1: is feature 1.
- w_0 or b: is the bias (the y-intercept).

Notethat

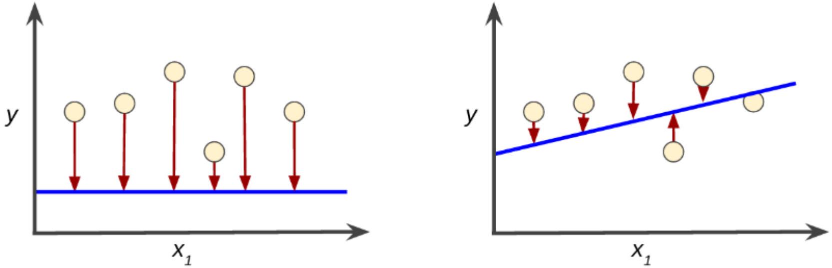
A model that relies on three features might look as follows:

$$y' = w_3 x_3 + w_2 x_2 + w_1 x_1 + w_0$$

Training and Loss

- **Training** a model simply means learning (determining) good values for all the weights and the bias from labeled examples.
- **Loss** is the penalty for a bad prediction.
    - Perfect prediction means the loss is zero
    - Bad model have large loss.
- Suppose we selected the following weights and biases.

Which of them have lower loss?



Squared loss

- The linear regression models use a popular loss function called squared loss.
- Also known as L_2.
- Is represented as follows:

$$[obsevation(x) - prediction(x)]^2$$
$$= (y - y')^2$$

Squared loss

- The linear regression models use a popular loss function called squared loss.
- Also known as L_2.
- Is represented as follows:

$$[obsevation(x) - prediction(x)]^2$$
$$= (y - y')^2$$

Why squared loss?

Squared loss

- The linear regression models use a popular loss function called squared loss.
- Also known as L_2.
- Is represented as follows:

$$[obsevation(x) - prediction(x)]^2$$
$$= (y - y')^2$$

Why squared loss?
Can we do absolute loss?

Mean square error (MSE)

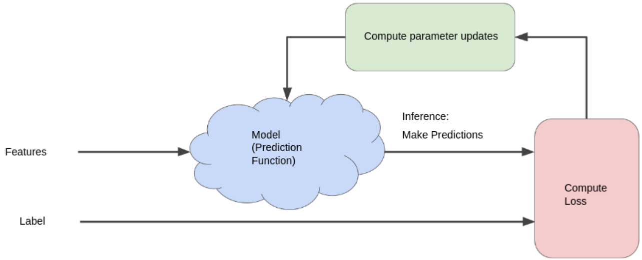
- Is the average squared loss per example over the whole dataset.

$$\text{MSE} = \frac{1}{N} \sum_{(x,y) \in D} (y - prediction(x))^2$$

- (x,y) is an example in which
  - y is the label
  - x is a feature
- prediction(x) is equal $y' = w_1 x + w_0$
- D is the dataset that contains all (x,y) pairs
- N is the number of samples in D

Reducing Loss

- **Training** is a feedback process that use the loss function to improve the model parameters.
- The **training** is an iterative process.



Two Questions

- What initial values should we set for w_1 and w_0?
- How to update w_1 and w_0?

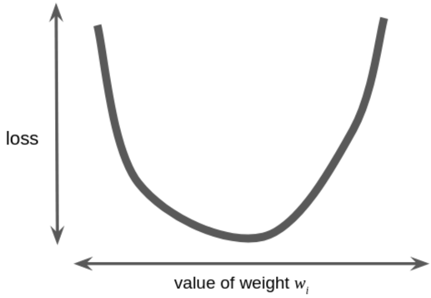
Gradient Descent (1/3)

- Assume (for symplicity) we are only concerned with finding w_1.
- Assume we had the time and the computing resources to calculate the loss for all possible values of w_1.

Gradient Descent (1/3)

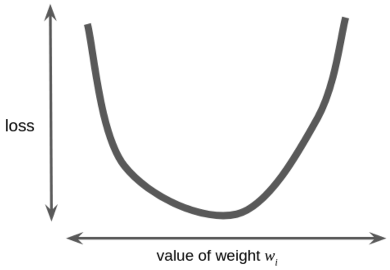
- Assume (for symplicity) we are only concerned with finding w_1.
- Assume we had the time and the computing resources to calculate the loss for all possible values of w_1.

Regression problems yield convex loss vs. weight plots.



Gradient Descent (2/3)

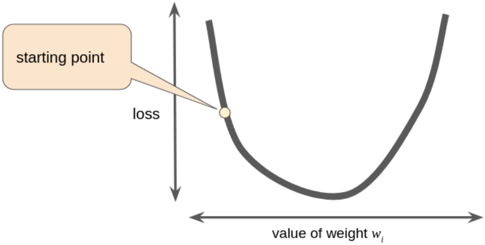
- Gradient descent enables you to find the optimal w without computing for all possible values.
- Gradient descent has the following steps
    1. Pick a random starting point for w
    2. Calculates the gradient of the loss curve at w.
    3. Update w
    4. go to 2, till convergence



Gradient Descent (3/3)

Note that a gradient is a vector, so it has both of the following characteristics:

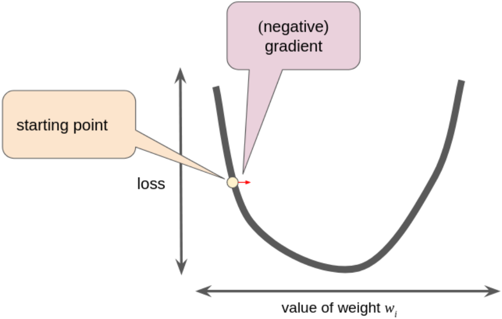
- Magnitude
- Direction



$$w_{new} = w_{old} - \eta * \frac{d \ \ loss}{dw}$$

Gradient Descent (3/3)

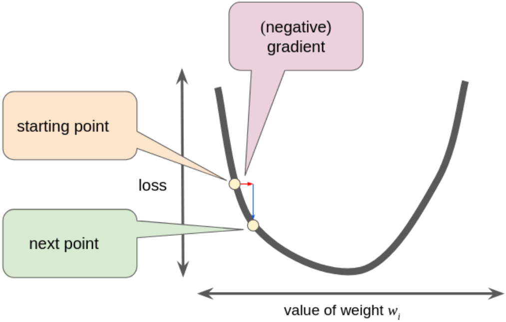
The gradient descent algorithm takes a step in the direction of the negative gradient



$$w_{new} = w_{old} - \eta * \frac{d\ \ loss}{dw}$$

Gradient Descent (3/3)

the gradient descent algorithm **adds some fraction of the gradient's magnitude (Learning Rate η)** to the previous point



$$w_{new} = w_{old} - \eta * \frac{d\ loss}{dw}$$

Convergence Criteria

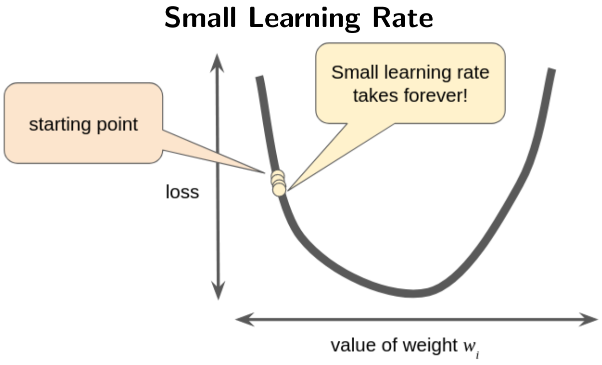
- For convex functions, optimum occurs when
  - $|\frac{d \ loss}{dw}| = 0$
- In practice, stop when
  - $|\frac{d \ loss}{dw}| \leq \epsilon$

Learning rate

- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?

Learning rate

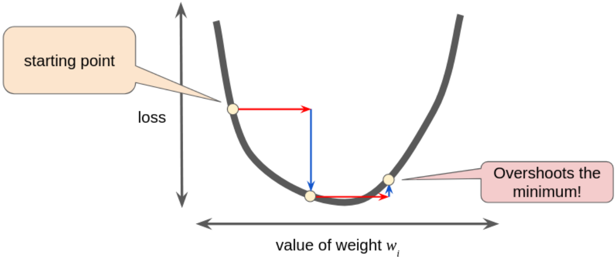
- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?



Learning rate

- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?

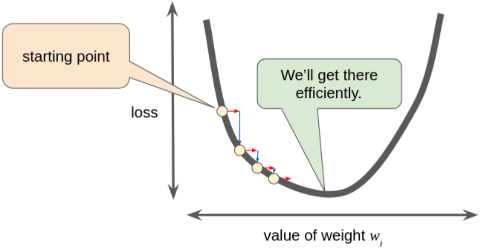
Large Learning Rate



Learning rate

- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?

Optimal Learning Rate usually (0.01)



Generalization and Gradient

- For n features: $y' = \sum_{i=0}^{i=n} w_i x_i$
- Note w_0 is the bias (intercept), and $x_0 = 1$.
- vector representation y' = $\mathbf{w}^T\mathbf{x}$
- Loss = $\ell = (y - y')^2$
- Gradient derivation

$$\begin{aligned}\frac{d\ell}{dw_i} &= \frac{d\ell}{dy'}\frac{dy'}{dw_i} \\ &= [2(y - y') * x_i * (-1)]\end{aligned}$$

Types of Gradient Descents

- **Batch Gradient Descent:**
  - MSE loss assumes taking gradient for the total number of samples in the data set
  - Data sets often contain billions or even hundreds of billions of examples
  - Can take a very long time to compute.
- **Stochastic Gradient Descent (SGD):**
  - Uses only a single example (a batch size of 1) per iteration.
  - Very noisy.
- **Mini-Batch Gradient Descent:**
  - Compromise between full-batch iteration and SGD
  - Typically a batch of size between 10 and 1,000 examples, chosen at random.

Questions