Machine Learning Overview and Context

Justin Silverman

01/19/2021

Table of Contents

What is Machine Learning?

"Field of study that gives computers the ability to learn without being explicitly programmed" – Arthur Samuel (1959)

We typically think of training and testing. Training is where the computer is training to do a task, Testing is where we are asking the computer to do the task.

Bad Machine Learning models don't generalize beyond the training data.

Section 1

Overview of ML problems

Types of ML Problems

Pretend that you want to sell your old cellphone on craigslist. In preparation, you have collected the following data for 1000 prior sales:

- Ultimate sale price *(how much money the seller ultimately made)*
- Listing price *(how much money the seller asked for initially)*
- Phone model
- Phone condition

Types of ML Problems

Pretend that you want to sell your old cellphone on craigslist. In preparation, you have collected the following data for 1000 prior sales:

- Ultimate sale price *(how much money the seller ultimately made)*
- Listing price *(how much money the seller asked for initially)*
- Phone model
- Phone condition

Prediction If I list my phone for $300 (and given its model and condition), how much can I expect it to sell for?

Types of ML Problems

Pretend that you want to sell your old cellphone on craigslist. In preparation, you have collected the following data for 1000 prior sales:

- Ultimate sale price *(how much money the seller ultimately made)*
- Listing price *(how much money the seller asked for initially)*
- Phone model
- Phone condition

Prediction If I list my phone for $300 (and given its model and condition), how much can I expect it to sell for?

Inference How does listing price effect the ultimate sale price? Is this relationship the same for all phone models/conditions?

Types of ML Problems

Pretend that you want to sell your old cellphone on craigslist. In preparation, you have collected the following data for 1000 prior sales:

- Ultimate sale price *(how much money the seller ultimately made)*
- Listing price *(how much money the seller asked for initially)*
- Phone model
- Phone condition

Prediction If I list my phone for $300 (and given its model and condition), how much can I expect it to sell for?

Inference How does listing price effect the ultimate sale price? Is this relationship the same for all phone models/conditions?

Decision Given that I ultimately want to make $300 for my phone, how much should I list it for?

Section 2

Overview of ML learning strategies

Types of ML learning strategies

As an example consider we want a computer to predict Y from X.

Supervised Learning (Most common) Learn how to predict Y from X given training examples that where both Y and X were known.

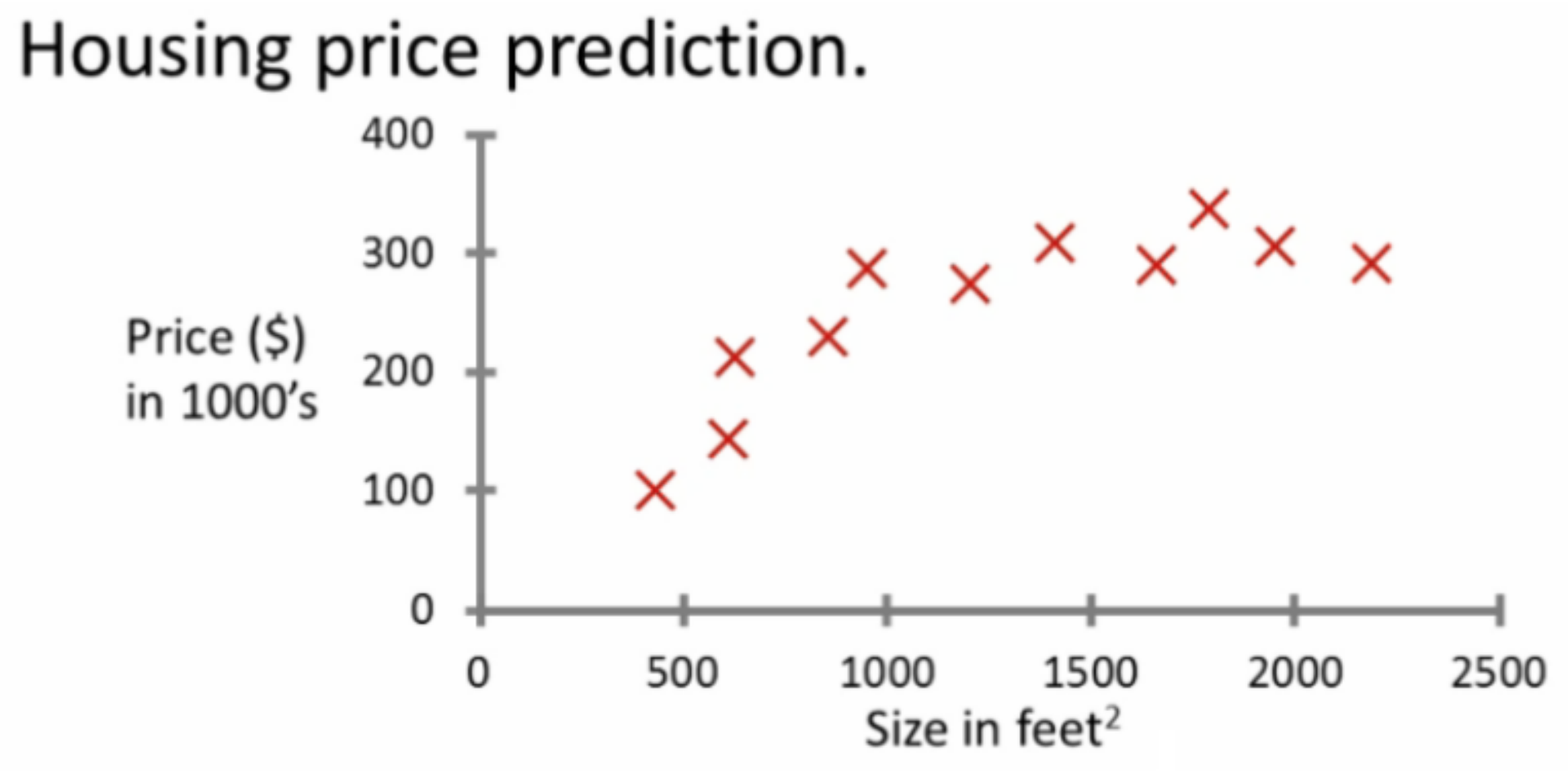
Types of ML learning strategies

As an example consider we want a computer to predict Y from X.

Supervised Learning (Most common) Learn how to predict Y from X given training examples that where both Y and X were known.

Unsupervised Learning (Also fairly common) Learn how to predict Y from X given training examples where only X is known.

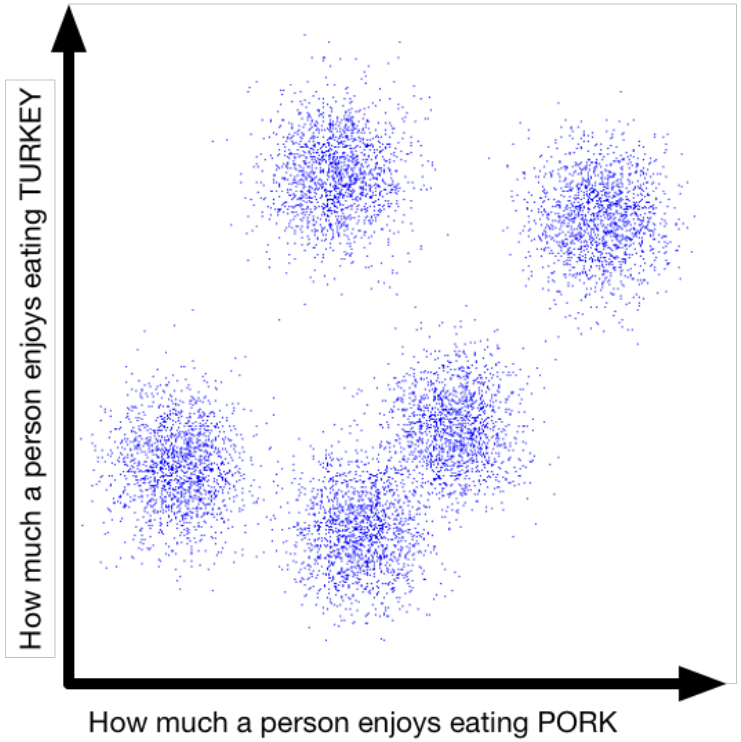
Supervised Learning (Prediction Example)



Given this data, a friend has a house 750 square feet - how much can they be expected to get? [1]

[1] Figure and example adopted from Andrew Ng – Machine Learning

Unsupervised Learning (Inference Example)



Label each data-point into 1 of 5 clusters.[2]

[2]Figure modified from David Sontag – NYU Clustering Lecture

Types of ML learning strategies

As an example consider we want a computer to predict Y from X.

Supervised Learning (Most common) Learn how to predict Y from X given training examples that where both Y and X were known.

Unsupervised Learning (Also fairly common) Learn how to predict Y from X given training examples where only X is known.

Types of ML learning strategies

As an example consider we want a computer to predict Y from X.

Supervised Learning (Most common) Learn how to predict Y from X given training examples that where both Y and X were known.

Unsupervised Learning (Also fairly common) Learn how to predict Y from X given training examples where only X is known.

Semi-Supervised (Less common) Learn how to predict Y from X given training examples where X is known for all training examples but Y is only known for a subset of the training examples.

Types of ML learning strategies

As an example consider we want a computer to predict Y from X.

Supervised Learning (Most common) Learn how to predict Y from X given training examples that where both Y and X were known.

Unsupervised Learning (Also fairly common) Learn how to predict Y from X given training examples where only X is known.

Semi-Supervised (Less common) Learn how to predict Y from X given training examples where X is known for all training examples but Y is only known for a subset of the training examples.

Reinforcement Learning (Less common) Learn how to predict Y from X where you (the Oracle) known Y and X but you only let the computer know X – the computer gets rewarded / punished based on how accurate/inaccurate its guesses for Y are.

Types of ML learning strategies

As an example consider we want a computer to predict Y from X.

Supervised Learning (Most common) Learn how to predict Y from X given training examples that where both Y and X were known.

Unsupervised Learning (Also fairly common) Learn how to predict Y from X given training examples where only X is known.

Semi-Supervised (Less common) Learn how to predict Y from X given training examples where X is known for all training examples but Y is only known for a subset of the training examples.

Reinforcement Learning (Less common) Learn how to predict Y from X where you (the Oracle) known Y and X but you only let the computer know X – the computer gets rewarded / punished based on how accurate/inaccurate its guesses for Y are.

Active Learning (Less Common) Learn how to predict Y from X given many examples of X but it is costly to label Y. So you ask the computer to figure out which examples would be the most

Terminology and Notation

Y aka: Labels, Dependent Variable
think "output"

X aka: Features, Independent Variables, Covariates
think "input"

N the number of data points/samples

P the number of parameters in a model or the number of features

e.g., $X_{n,p}$ refers to the p-th feature in the n-th sample

e.g., Y_n refers to the label of the n-th sample

Section 3

Another Categorization of ML Problems

More detailed overview of ML Problems

- Regression
- Classification
- Clustering
- Anomaly Detection
- Feature Selection
- Dimensionality Reduction

Regression

- **Y is continuous** (e.g., a dollar amount)
- X can be anything (continuous, discrete, etc...)
- typically a type of supervised learning

Example Models:

- Linear Regression
- Generalized Linear Regression
- Gaussian Process Regression (Non-linear)
- Basis Function / Spline Models (Non-Linear)
- Neural Networks (Non-linear)

Classification

- **Y is discrete** (e.g., disease vs. health)
- X can be anything (continuous, discrete, etc...)
- typically a type of supervised learning

Example Models:

- Logistic Regression (Linear Classifier, 2 class labels)
- Categorical (aka softmax) Regression (Linear Classifier, 2 or more class labels)
- Gaussian Process Classification (Non-Linear classifier, 2 or more class labels)
- Neural Networks (Non-Linear classifier, 2 or more class labels)
- Naive Bayes (Linear classifier, 2 or more class labels)
- Nearest Neighbor Classification (Non-Linear classifier, 2 or more class labels)

Clustering

- Y is discrete
- X can be anything (continuous, discrete, etc...)
- typically a type of **unsupervised learning**

Example Models:

- K Nearest Neighbors
- Mixture Models
- Hierarchical Clustering

Anomaly Detection

- No Y
- Given a bunch of examples of X, find the subset of examples that are "anomalous", e.g., outliers.
- typically a type of **unsupervised learning**
- Example: Identify potential fraudulent credit card purchases

Example Models – we will learn a number of them but its hard to give them discrete names.

Feature Selection (aka Variable Selection)

- Supervised or Unsupervised
- (Supervised) Figure out which feature(s) $X_1, \ldots, X_P$ are the most useful in relating Y and X.
- (Unsupervised) Figure out which feature(s) $X_1, \ldots, X_P$ distinguish the data points.
- Supervised example: Which 1 feature (phone model vs. phone condition) is most important in predicting ultimate sales price?

Example Models:

- Lasso and Other Penalized Regression
- Forward / Backwards Selection

Dimensionality Reduction

- Supervised or (more commonly) Unsupervised
- (Unsupervised) Figure out new features $Z_1, \ldots, Z_K$ that are some function of $X_1, \ldots, X_P$ and such that $K << P$ but Z still maintains most of the variation/signal in the original data.
- This is often used to visualize high-dimensional data and see patterns

Example Models:

- Principal Component Analysis
- Non-metric Dimensional Scaling
- Auto-Encoders (type of Neural Network Model)

Section 4

Model Representations

Model Representations (Linear Regression Example)

Data never falls perfectly on a line! So we need to expand the model for a simple line ($y = mx + b$) to include error.

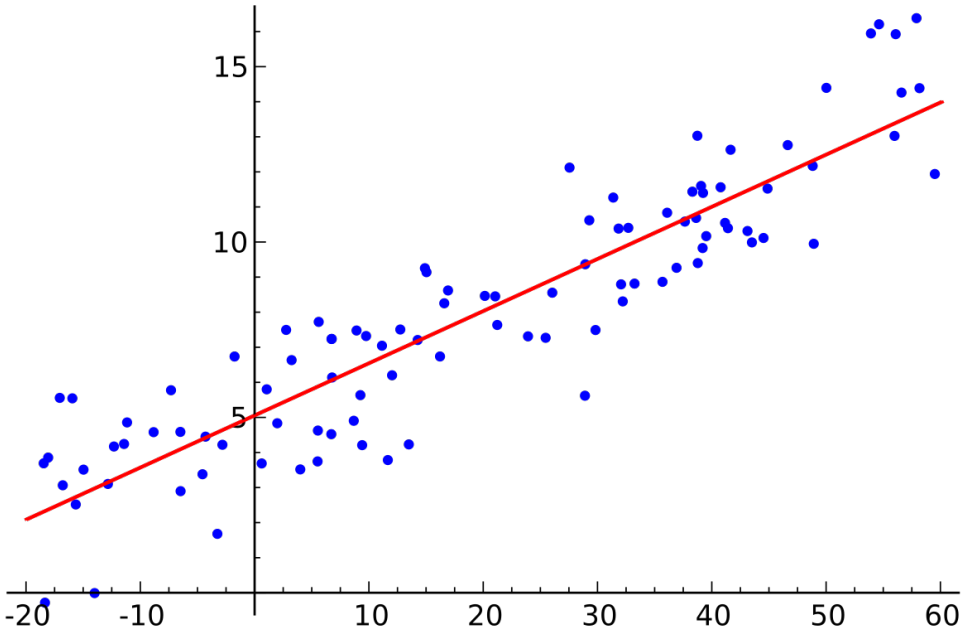


Figure 1: (Image from Wikipedia)

Model Representations (Linear Regression Example)

All models have multiple representations. The following two representations of Linear Regression are completely equivalent and result in the same estimate for β.

Probabilistic Representation of Linear Regression

$$\beta = \underset{\beta,\sigma^2}{\operatorname{argmax}} \prod_i N(Y_i|\beta X_i, \sigma^2)$$

Loss Representation of Linear Regression

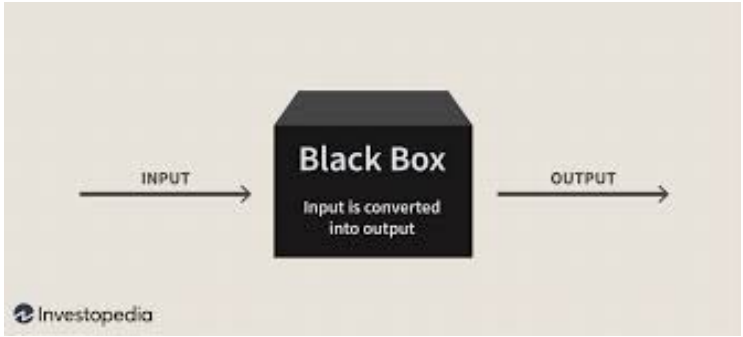
$$\beta = \underset{\beta,\sigma^2}{\operatorname{argmin}} \sum_i (X_i\beta - Y_i)^2$$

Section 5

What are "Black Box Models"

What are "Black Box Models"

Sometimes we care only about predictive performance without caring about "modeling" a realistic process. In such cases we may turn to "Black Box Models." You may not care whats in the black box if you only care about predictive performance (*e.g.*, minimizing misclassification error).



Warning!

Black Box models can be powerful, but there are times where they will fail to generalize beyond the training set as they may learn patterns that are completely unrealistic.

Section 6

Model Evaluation

Model Evaluation Topics

Key questions we are trying to answer here:

- How do you pick model hyper-parameters?
- When should we trust our models on new data and when should we not?
- How do we teach a computer to build better models? How do we determine if one model is better than another?

Section 7

The $P >> N$ Problem

The $P >> N$ Problem

A central challenge in Machine Learning is that there are often more model parameters (P) than there are data-points (N). This can lead to non-identifiability in many models: **There is no longer a single solution to the ML problem but a family of equally likely solutions that cannot be distinguished based on the limited data available.**

The $P >> N$ Problem

A central challenge in Machine Learning is that there are often more model parameters (P) than there are data-points (N). This can lead to non-identifiability in many models: **There is no longer a single solution to the ML problem but a family of equally likely solutions that cannot be distinguished based on the limited data available.**

Example

You are a biologist trying to predict the height an individual i based on their genome. Your data-set consists of 1,000 individuals and 20,000 binary genetic markers (e.g., X_{ij}) represents whether the i-th individual has ($X_{ij} = 1$) or does not have ($X_{ij} = 0$) the j-th genetic marker. You decide to use linear regression:

$$Y_i = \beta_1 X_{i1} + \cdots + \beta_{20000} X_{i\,20000} + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

You have a problem here.

Common Approaches to Solving $P >> N$ Problem

- Penalization - e.g., Lasso, Ridge, or Elastic Net Regression
- Bayesian Regression

Example

(continued from prior slide) The prior model can be written in its loss formulation as:

$$\beta_1, \ldots, \beta_{20000} = \underset{\beta_1,\ldots,\beta_{20000}}{\operatorname{argmin}} \sum_i \left(Y_i - \sum_{j=0}^{20000} \beta_j X_{ij} \right)^2 .$$

If we are willing to assume that on average the genetic markers have a small effect then there exists a unique solution. Ridge Regression is one solution to the $P >> N$ problem that reflects this assumption.

$$\beta_1, \ldots, \beta_{20000} = \underset{\beta_1,\ldots,\beta_{20000}}{\operatorname{argmin}} \sum_i \left(Y_i - \sum_{j=0}^{20000} \beta_j X_{ij} \right)^2 + \sum_j \beta_i^2$$

Section 8

Machine Learning vs. Statistics

Machine Learning vs. Statistics

In truth they are identical, they are both concerned with learning from data. That said, in practice the two fields have different focuses and it is useful to think broadly about the practical differences.

- Statistics tends to focus on Inference whereas Machine Learning tends to focus on Prediction.
- You will rarely see hypothesis testing (e.g., p-values) in Machine learning.
- Black-box models are far more common in machine learning. ML has more of a "who cares as long as it works" mentality.
- Machine Learning tends to specify models in terms of their loss-function representation whereas Statistics tends to specify models based on their probabilistic representations. (That said, probabilistic machine learning is a very active field at the moment.)

A Word of Caution

Machine Learning can be incredibly powerful and has revolutionized many fields. At times ML can feel like magic.

But it has limitations that must be respected. **Always remember that most of these algorithms are guaranteed to give you an answer.** Given properly formatted data they will give you a prediction. But that doesn't mean you should trust the prediction.

A Word of Caution

Machine Learning can be incredibly powerful and has revolutionized many fields. At times ML can feel like magic.

But it has limitations that must be respected. **Always remember that most of these algorithms are guaranteed to give you an answer.** Given properly formatted data they will give you a prediction. But that doesn't mean you should trust the prediction.

Machine learning can fail spectacularly. You can do unbelievably stupid things with ML and **it can be incredibly difficult to realize there is a problem.**

A Word of Caution

Machine Learning can be incredibly powerful and has revolutionized many fields. At times ML can feel like magic.

But it has limitations that must be respected. **Always remember that most of these algorithms are guaranteed to give you an answer.** Given properly formatted data they will give you a prediction. But that doesn't mean you should trust the prediction.

Machine learning can fail spectacularly. You can do unbelievably stupid things with ML and **it can be incredibly difficult to realize there is a problem.**

ML can also do unexpected things. Think of it like a child who misunderstands your teaching in unexpected ways.

Always be particularly skeptical when using ML for inference.

Section 9

Types of Data / Data Representations

Discrete vs. Continuous

These are broad terms that you should be familiar with:

Continuous Think of data that can have decimals. (e.g., height)

Discrete Think of data that takes on discrete categories / values (e.g., number of people in this course or whether someone has a pet dog).

Discrete vs. Continuous

These are broad terms that you should be familiar with:

Continuous Think of data that can have decimals. (e.g., height)

Discrete Think of data that takes on discrete categories / values (e.g., number of people in this course or whether someone has a pet dog).

A note on data representations

What follows is an introduction to different types of data. Some of these categories are mutually exclusive (e.g., real vs. discrete data) but others are not. Consider that you can have a time-series of compositions or a time-series of counts.

Real Valued Data

We write $x \in \mathcal{R}$, *i.e.*, data that can take on any value in the interval $(-\infty, \infty)$. Multivariate real valued data is $x \in \mathcal{R}^D$ where D is the dimension of the data.

- Example: how much weight a person has gained or lost
- These data are easy to represent in a model / on a computer – *e.g., a vector of doubles in C++*

Positive Valued Data

We write $x \in \mathcal{R}^+$, *i.e.*, data that can take on any value in the interval $[0, \infty)$ or sometimes we exclude zero and write $(0, \infty)$. Multivariate positive valued data can be written as $x \in \mathcal{R}^{D+}$ (*i.e.* these data exist in the positive orthant of D dimensional real space).

- Example: A persons weight or height
- Most measurements of real quantities are positive.
- These data are easy to represent in a model / on a computer – *e.g.*, *exponentiated vector of doubles in C++*.

Count Data

We write $x \in \mathcal{Z}$, *i.e.*, data that can take on positive integer value $\{0, 1, 2, 3 \dots\}$. Multivariate count data can be represented as $x \in \mathcal{Z}^D$.

- Univariate Example: The number of students enrolled a given at the university
- Multivariate Example: The number of D different species of tree present in a given plot of land
- These data are easy to represent in a model / on a computer – *e.g.*, *a vector integers*

Discrete Data with 2 Categorical levels

We write $x \in \{0, 1\}$, *e.g.*, data that can be represented as "yes/no" or "true/false".

- Example: Does a person have a disease or not (e.g., Alzheimer's disease vs. No Alzheimer's disease)
- These data are easy to represent in a model / on a computer – *e.g., a boolean*

Discrete Data with more than 2 categorical levels

We write $x \in \{0, 1, \ldots, K\}$, *e.g.*, data that can be represented as one of K different categorical levels.

- Example: University roles (e.g., Faculty vs. Undergraduate Student vs. Graduate Student vs. Post-doc, etc...)
- These data typically require some thought when including them in models (see example).

Discrete Data with more than 2 categorical levels

Example

Let y be an outcome of interest (e.g., height) and x be a categorical variable that can be one of 3 distinct categories (e.g., whether a persons favorite color is red, blue, or yellow).

How would we regress y against x using a linear model?

Encoding $x = 0$ if "red", $x = 1$ if "blue", and $x = 2$ if "yellow" makes no sense as it implies that "yellow" effects y twice as much as "blue" (think $y = \beta x$ in this case). Two common options are the one-hot-encoding and the dummy-variable encodings.

Encodings for Discrete Data with more than 2 (unordered) categorical levels}

For discrete data $x \in \{1, \ldots, K\}$

One-Hot Encoding Create a new variable z that is categorical with 2 levels but D dimensions. That is let $z \in \{0, 1\}^K$. We write $z = \{z_1, \ldots, z_K\}$ and specify that $z_k = 1$ if $x = k$ and 0 otherwise.

Dummy Encoding Create a new variable z that is categorical with 2 levels but $K - 1$ dimensions. That is let $z \in \{0, 1\}^{K-1}$. We must choose a "reference" category (for example category K). We write $z = \{z_1, \ldots, z_{K-1}\}$. But here it gets a little tricky. We are going to encode the case where $x = K$ as $z = \{0, \ldots, 0\}$ (*i.e.*, all zeros). For the other $K - 1$ categories we specify that $z_k = 1$ if $x = k$ and 0 otherwise (just as in the one-hot encoding). In other words we are just dropping one dimension of z by encoding the reference category as the special case where all z_k are zero.

Ordinal Data

We say that x is ordinal if x takes on discrete values in the set $\{1, \ldots, K\}$ where the values $\{1, \ldots, K\}$ have a natural ordering to them.

- Contrast this with the prior definition which was ordered.
- Example: Shirt sizes (Small, Medium, Large)
- Encodings are challenging and are often problem specific. Consider that encoding (small=0, medium=1, large=2) assumes again that large is 2 times larger than medium. Often this is too strong an assumption.
- These data often require special models.

Interval Data

We say that x is an interval measurement if it can be written as a discrete or continuous set (high,low) where the boundary elements can also be in the set. For example, [high,low) is also interval data.

- Example: Survey data where ages have been binned for anonymity, *i.e.*, 0-4 years, 5-17-years, . . . , 65+ years.
- Encoding are challenging and often problem specific.
- Often requires specialized models

Censored Data

We say that x is censored if certain values of x cannot be measured or are grouped together. We often refer to data as either being right (high values) or left (low values) censored Its best to understand this by examples.

- **Example of left censored data:** Measurements of a trace chemical in drinking water where your measurement technology can only detect the chemical if it is above a certain concentration (*i.e.*, above the limit of detection). You can imagine that even if the chemical is present but below the limit of detection then your measurement just comes back as zero.
- **Example of right censored data:** Same as before (you are measuring a trace chemical in drinking water) but now your measurement technology saturates above a certain threshold (*e.g.*, all concentrations 1000 parts per million come back as 1000 parts per million).
- **Another example of right censored data:** Ever try to take a picture of the sun? Notice how you just get a white image – the image saturates and above a certain threshold you just get a totally white image.

Censored Data (Continued)

- You can have data that is both right and left censored.
- This data requires special techniques to model

Time to Event Data

Just as the name implies, data where you are measuring a time to some event. Again, best to describe through examples:

- Example: You are running a clinical trial for a new drug that is supposed to decrease the risk of heart attack in patients. In the treatment and control group of your trial your outcome of interest is the time to heart attack since the start of the trial. Consider that you get censored data here too, *e.g.*, what if a patient dies in a car accident (would they have gone on to have a heart attack if they had not died of a different cause?).
- These data require special models (e.g., Cox Proportional Hazard models)

Time Series Data

Sometimes also called "Longitudinal Data". We say x is a time-series if it is a vector $x = (x_1, \ldots, x_T)$ of repeated measurements that are ordered in time.

- Example: Financial data (e.g., stock prices) are often analyzed as time-series
- Example: A persons weight over time
- These data can be regularly spaced (e.g., measurements are taken ever day) or irregularly spaced (e.g., measurements are taken whenever a patient shows up to the doctor) irregularly spaced data requires special consideration and not all models can handle this type of data.
- You can have both continuous time-series or discrete time-series. *e.g.*, you can have a time-series of counts.
- All time-series typically require special models
- All time-series are multivariate in the sense that they involve multiple measurements over time. Still, the convention is to call a time-series multivariate only if, at ever time point, a multivariate measurement is collected (e.g., 5 stocks are being tracked over time).

Functional Data

Where your data is random functions, *e.g.*, x if it can be represented as $y = x(z)$.

- Consider that this data is actually infinite dimensional! Think about every possible function relating y and z!
- There is a lot of overlap between time-series and functional data.
- This data often requires special models and techniques.
- Encodings are often challenging but there are good tools and encodings avaialble.

Compositional Data

We write $x \in \mathcal{S}^D$ (e.g., the D-dimensional simplex) and call x a composition if

$$x = (x_1, \ldots, x_D) : \left\{ \sum_i x_i = k \text{ and } x_i \in \mathcal{R}^+ \text{ for all } i \right\}.$$

- Examples: Proportions, Percentages, parts per million, anything that is a positive measurement that sums to a constant value.
- Example: How people spend their days, *i.e.*, the proportion of time in each day that you sleep, eat, and study (there are only 24 hours in the day).
- Central to this data is the competition between the x_i, *e.g.*, if you spend more of your day sleeping you have less time to study + eat.
- This data is challenging to work with and requires special encodings. *e.g.*, additive log-ratio transform where we encode x as a new variable y defined as $(y_1, \ldots, y_{D-1}) = \left(\log \frac{x_1}{x_D}, \ldots, \log \frac{x_{D-1}}{x_D} \right)$.

Lattice Valued Data

Also known as Compositional Count Data. Lattice valued data is like compositional data but the values are constrained to be integer values.

$$x = (x_1, \ldots, x_D) : \left\{ \sum_i x_i = k \text{ and } x_i \in \mathcal{Z}^+ \text{ for all } i \right\}.$$

Example: You have a giant ball-pit with red, green, and blue balls. The ball pit is too big to count all the balls so instead you reach in and grab an armful and just count those. The number of balls you can pick up is limited by the length of your arms. This data occurs frequently in surveys or in bioinformatics / computational biology (e.g., microbiome data or gene expression data) * Example: You are polling 1000 individuals to predict who will win the next presidential election.

Text Data

There are lots of kinds of text data and many different encoding of text data.

- Example: wikipedia
- Example: doctors notes in an electronic hospital record
- One common encoding is the "bag of words encoding" where each word in the language is given a unique category and then the data is analyzed as unordered categorical data.
- Another encoding is the n-gram encoding where sets of words are treated as unorderd categorical data.
- There are many many other encodings.

Color Data

Think of the color of an individual pixel in an image. This data often exists in a cube or hyper-cube.

- Example encoding: RBG color scheme represents colors as triplets (x_1, x_2, x_3) where each x_i is a value between 0 and 255.

Image Data

There are lots of kinds of image data and many different encoding.

- Example: MRI data - 3 dimensional images made up of many 2 dimensional slides (greyscale)
- Example: Cell phone images (RGB encoded 2 dimensional array of pixels). Consider that for a 2000 by 2000 pixel image there are $(2000 \times 2000)^3$ dimensions in the RGB encoding.
- Example: Vector PDF images that are actually a set of functions rather than pixels.
- There are many encodings of images. A particularly popular one recently is the convolutions filters used in many Deep Neural Network models.
- Translational, scaling, and rotational invariance are big challenges in image data. Consider that in two images of cows, the cows may be at different distances from the camera, facing different directions, centered in a different part of the image. The images may also have different resolutions (one may be 2000 by 2000 the other may be 1000 by 2000).

Sequence Data

We call x a sequence if $x = (x_1, \ldots, x_S)$ where each $x_i \in \{1, \ldots, K\}$. e.g., an ordered set of categorical variables.

- You can often think of this data as a discrete, categorical, time series but where the sequence may not be collected over time.
- Example: DNA sequences where each x_i is a base (e.g., Adenine, Guanine, Cytosine, and Thymine).
- Example: The position of a robot moving around your home categorized by which room it is in: {Kitchen, Dining Room, Bedroom, Bathroom, Kitchen}.
- Typically requires special models. Encodings are often based on the same encodings used for categorical variables (e.g., one-hot or dummy encodings) but now we are dealing with ordered sets of these categorical encodings.

Directional Data

Typically either 2 or three dimensional and repesenting the direction an object is moving. This data can be thought to exist on a circle or a sphere.

- Example: the direction that a plane around the globe.
- This data can be challenging as the "start" and the "end" of any representation is often linked. For example, if you turn around 360 degrees you are back to where you started. This can make encodings challenging.

Network Data or Graph data

Thought of typically in terms of edges and verticies. This data can be directed or undirected. This data can be weighted or unweighted.

- Example: Contact networks (e.g., friend networks) on social media. Who knows who. Verticies are individuals who are liked by edges (edges represent the presence or absence of a connection between two individuals)
- Edges can have additional qualities such as weights, *e.g.*, how many messages are going between two individuals.
- Edges can be directed or undirected, *e.g.*, contacts on twitter are directed: You can follow anyone and they don't have to follow you back.
- Non-social network example: Rivers are directed networks with weights represnting the the amount of water flowing. Rivers are actually dynamic networks as both the connectivity and water flows change over time.

Shape Data

Best to understand thorugh examples:

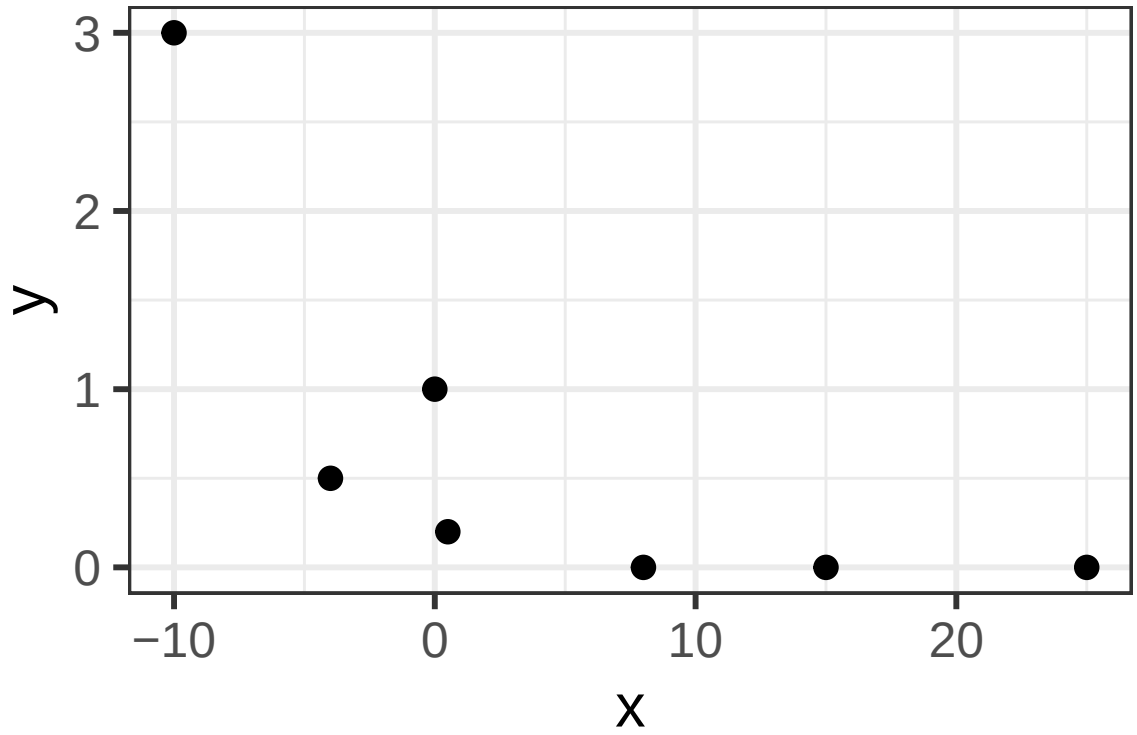
- Example: The shape of the teach of different primate species.
- Example: The shape of cars (e.g., you may be studying how visual aesthetics applied to automobiles have changed over time).
- Example: The shape/morphology of a brain tumor on MRI scans.
- This data is very challenging and can be thought of often as a very special type of functional data.
- There are many encoding and they are all fairly challenging.

Section 10

Data has Meaning, Your Choice of Encoding Matters

A simple example: Regresion with Positive Valued data.

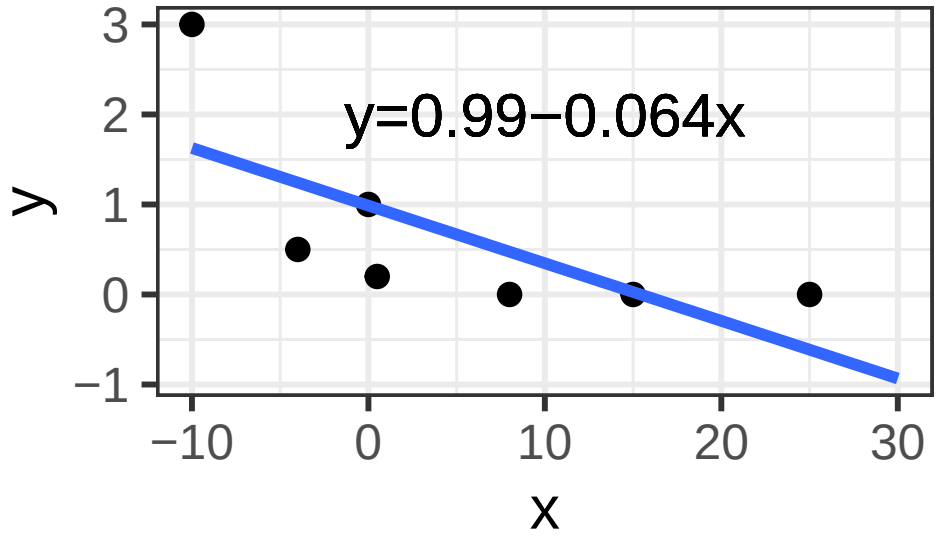
Joe is trying to model the amount of snow in his backyard as a function of the mid-day temperature.[3] Let y_i denote a measurement of the amount of snow at one spot in his backyard (measured in inches) and let x_i denote the mid-day temperature on the day measurement (in celsius) y_i was taken. Here is the data Joe collects:



[3] Lets present that these measurements are taken many days apart so we can safely ignore the fact that this is a time-series.

A simple example: Regresion with Positive Valued data.

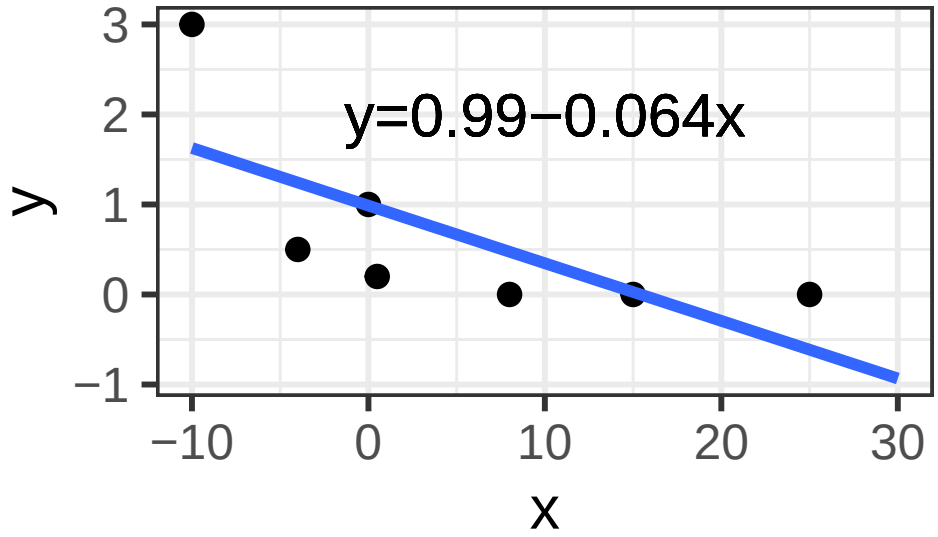
Joe decides to model this data with linear regression.



So here the model is saying that for every 1 degree decrease in temperature there is an additional -0.064 inches of snow.

A simple example: Regresion with Positive Valued data.

Joe decides to model this data with linear regression.



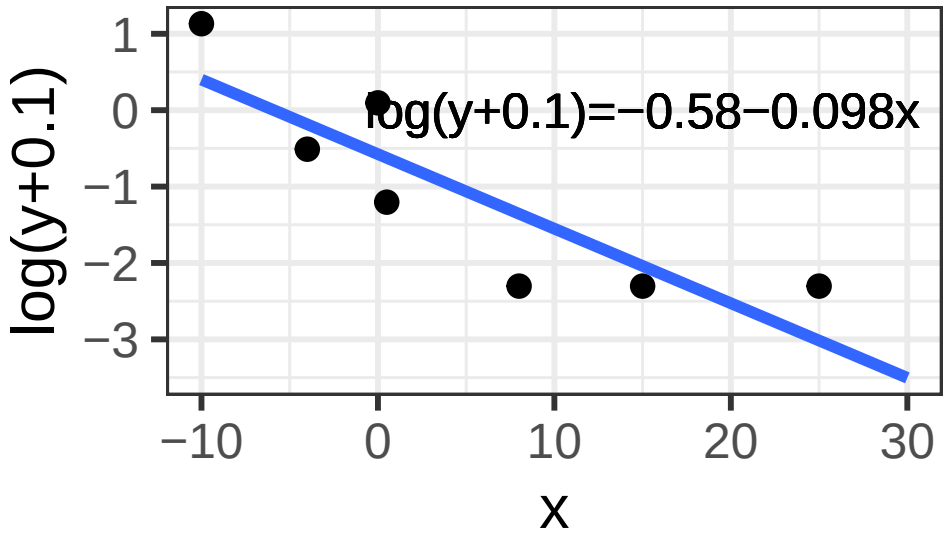
So here the model is saying that for every 1 degree decrease in temperature there is an additional -0.064 inches of snow.

But Wait!

This model predicts that above 15 degrees celsius, there is negative snow! Something must be wrong.

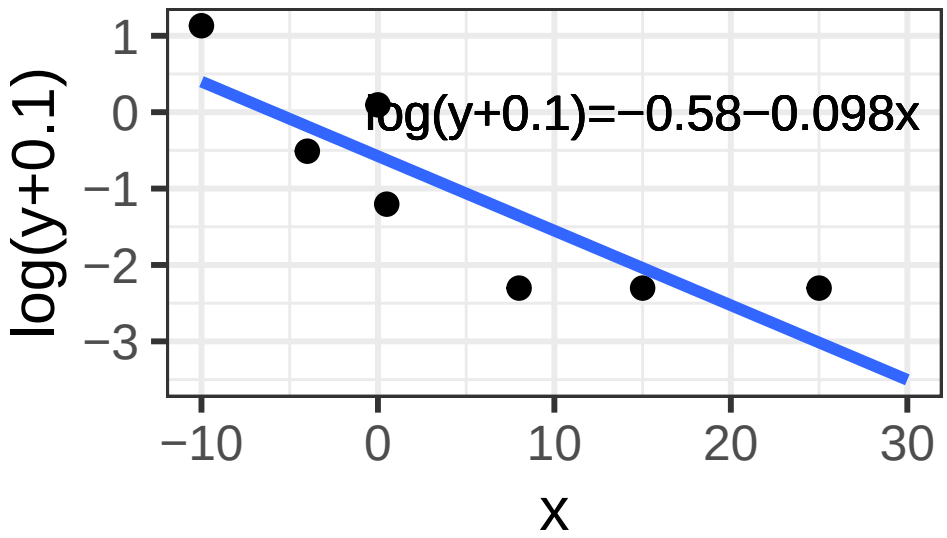
A simple example: Regresion with Positive Valued data.

To "fix" this problem Joe decides to log-transform the inches of snow (adding 0.1 to avoid negative infinity from zeros) and fit the regression to the log-transformed data.



A simple example: Regresion with Positive Valued data.

To "fix" this problem Joe decides to log-transform the inches of snow (adding 0.1 to avoid negative infinity from zeros) and fit the regression to the log-transformed data.

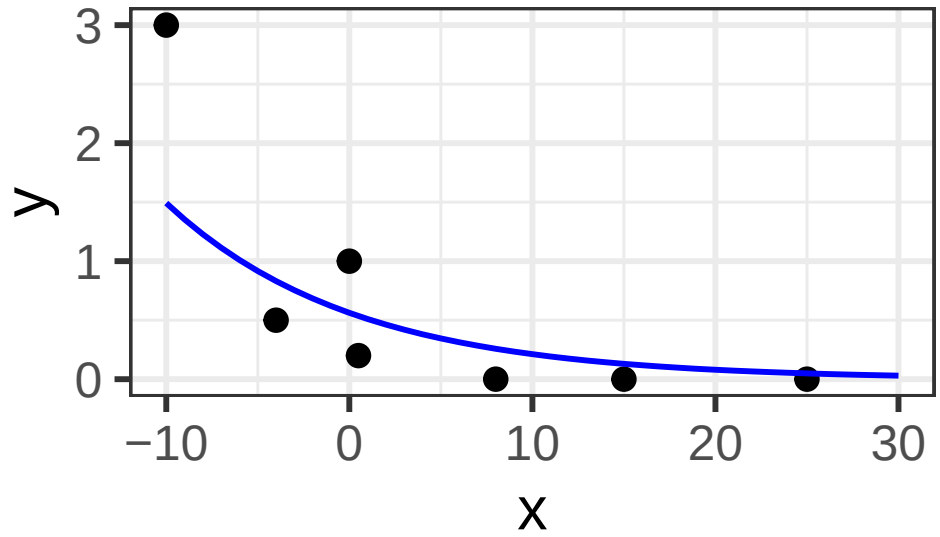


But Wait!

Now how do we interpret this model? Now the model says that for every 1 degree decrease the $log(y + 0.1)$ increases by -0.098. We are acctually now modeling that temperature has a multiplicative effect on snow fall, not an additive effect.

A simple example: Regresion with Positive Valued data.

Note the log-transformed model can can instead be viewed as an equivalent exponential model. This follows from the fact that $\log(y + 0.1) = mx + b$ implies that $y = e^{mx+b} + 0.1$.



Moral of the Story

The lesson here is that data has meaning. You need to consider that meaning when picking a representation / encoding. Also different encodings imply different models.