

Machine, Data and Learning

Selected slides for lectures on ML Topic

Machine Learning

- Based on Chapter 1 of Python Machine Learning by Example by Yuxi Liu
- Scientific study of algorithms and statistical models that computer systems use
 - To perform a specific task effectively without using explicit instructions
 - Rely on patterns and inference instead.
- Involves
 - Building a **mathematical model** based on sample data, known as "training data" to make predictions or decisions
 - No explicit programming done to perform the task

Machine Learning

- Term coined around 1960
- Why learn ? Why not just hire enough programmers and code in rules ?
 - Lots of patterns for an activity/event
 - Events can be dynamic
 - **Data** is increasing exponentially
 - **Data** is also in various formats [Text, Audio, Video]
 - Higher quality **data** due to cheaper storage
- Can be broadly classified into three categories
 - Unsupervised, Supervised and Reinforcement learning

Unsupervised Learning

- Takes a set of data points that contains only inputs and finds structure in data E.g., Grouping or Clustering of data points
- **Marketing:** Finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records.
- **Biology:** Classification of plants and animals given their features.
- **Earthquake studies:** Clustering observed earthquake epicenters to identify dangerous zones.
- **World Wide Web:** Clustering weblog data to discover groups of similar access patterns.

Supervised Learning

- Builds mathematical model using data set that has both inputs and desired outputs E.g., Classification and Regression tasks

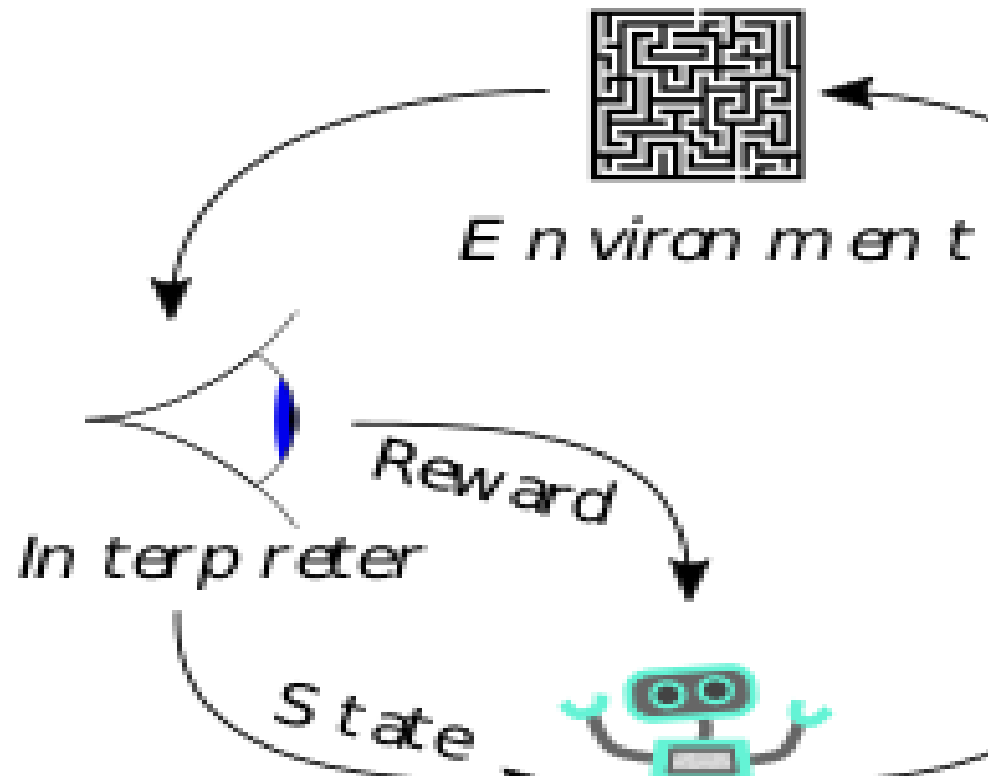
User ID	Gender	Age	Salary	Purchased	Temperature	Pressure	Relative Humidity	Wind Direction	Wind Speed
15624510	Male	19	19000	0	10.69261758	986.882019	54.19337313	195.7150879	3.278597116
15810944	Male	35	20000	1	13.59184184	987.8729248	48.0648859	189.2951202	2.909167767
15668575	Female	26	43000	0	17.70494885	988.1119385	39.11965597	192.9273834	2.973036289
15603246	Female	27	57000	0	20.95430404	987.8500366	30.66273218	202.0752869	2.965289593
15804002	Male	19	76000	1	22.9278274	987.2833862	26.06723423	210.6589203	2.798230886
15728773	Male	27	58000	1	24.04233986	986.2907104	23.46918024	221.1188507	2.627005816
15598044	Female	27	84000	0	24.41475295	985.2338867	22.25082295	233.7911987	2.448749781
15694829	Female	32	150000	1	23.93361956	984.8914795	22.35178837	244.3504333	2.454271793
15600575	Male	25	33000	1	22.68800023	984.8461304	23.7538641	253.0864716	2.418341875
15727311	Female	35	65000	0	20.56425726	984.8380737	27.07867944	264.5071106	2.318677425
15570769	Female	26	80000	1	17.76400389	985.4262085	33.54900114	280.7827454	2.343950987
15606274	Female	26	52000	0	11.25680746	988.9386597	53.74139903	68.15406036	1.650191426
15746139	Male	20	86000	1	14.37810685	989.6819458	40.70884681	72.62069702	1.553469896
15704987	Male	32	18000	0	18.45114201	990.2960205	30.85038484	71.70604706	1.005017161
15628972	Male	18	82000	0	22.54895853	989.9562988	22.81738811	44.66042709	0.264133632
15697686	Male	29	80000	0	24.23155922	988.796875	19.74790765	318.3214111	0.329656571
15733883	Male	47	25000	1					

Figure A: CLASSIFICATION

Figure B: REGRESSION

Reinforcement Learning

- Concerned with how software agents should take actions in an environment to maximize cumulative reward E.g. Autonomous vehicles, Computer games



Some Applications

- Search engines
- Information retrieval
- Recommendation systems
- Credit card fraud detection
- Disease diagnosis
- Election prediction
- Image processing
- Speech translation
- ...

AlphaGo

- First computer Go program to defeat a 9-dan professional player
- Uses Monte Carlo Tree search algorithm based on knowledge learned by a deep learning method
- Beat World No. 1 ranked player in 2017
 - Retired after this match
- <https://deepmind.com/research/alphago/>
- <https://www.youtube.com/watch?v=WXuK6gekU1Y>
- AlphaGo Zero – Version without human data and stronger than AlphaGo [defeated 100-0]

AlphaZero & MuZero

- AlphaZero, a generalized version of AlphaGo Zero
Took 4 hours to learn Chess and defeat reigning world computer chess champion 28 to 0 in 100 matches
- <https://www.youtube.com/watch?v=WXHFqTvfFSw>
- MuZero: Master games without knowing rules
- Uses approach similar to AlphaZero, developed in 2019
- Trained via self-play and play against AlphaZero with no access to rules, opening books or endgame tables
- Viewed as significant advancement over AlphaZero

AlphaFold: solution to a 50 year old grand challenge in biology

- <https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology>
- Figuring out what shapes proteins fold into is known as the “protein folding problem” - grand challenge in biology for the past 50 years
- Focus of intensive scientific research for many years, using a variety of experimental techniques such as nuclear magnetic resonance and X-ray crystallography.

AlphaFold

- Number of ways a protein could theoretically fold before settling into its final 3D structure is astronomical.
- Cyrus Levinthal estimated 10^{300} possible conformations for a typical protein.
- Estimated would take longer than the age of universe to enumerate all possible configurations. Yet in nature, proteins fold spontaneously, some within milliseconds - referred to as Levinthal's paradox.

Generalization & Goodness of Fit

- **Generalization** refers to how well the concepts learned by a ML model generalizes to specific examples or data not yet seen by the model.
 - ...
- **Goodness of fit** describes how well a model fits for a set of observations.
 - Overfitting and Underfitting

Overfitting

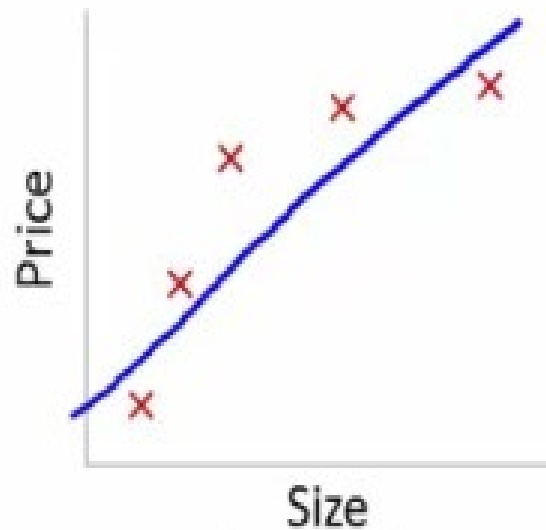
- Phenomenon of extracting too much information from training sets or memorization can cause overfitting
 - Makes ML model work well with training data called **low bias**
 - Bias refers to error due to incorrect assumptions in learning algorithm
 - However, does not generalize well or derive patterns, performs poorly on test datasets called **high variance**
 - Variance measures error due to small fluctuations in training set

Underfitting

- Model is underfit if it does not perform well on training sets and will not do so on test sets
- Occurs when we are not using enough data to train or if we try to fit wrong model to the data
 - E.g., if you do not read enough material for exam or if you prepare wrong syllabus
- Called **high bias** in ML although **variance is low** [i.e. consistent but in a bad way]
- May need to increase number of features since it expands the hypothesis space.

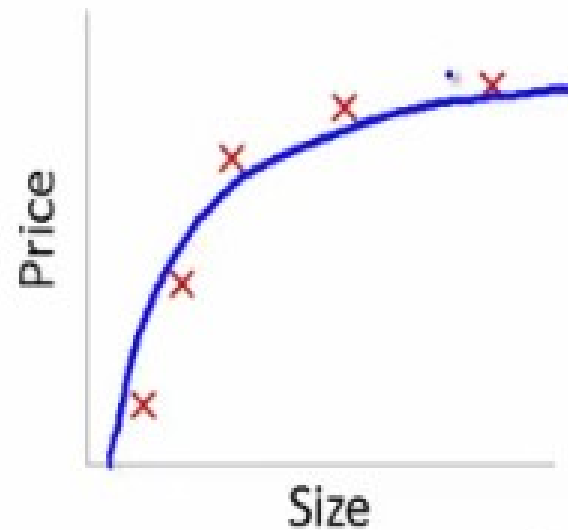
Goodness of fit

- For same data:



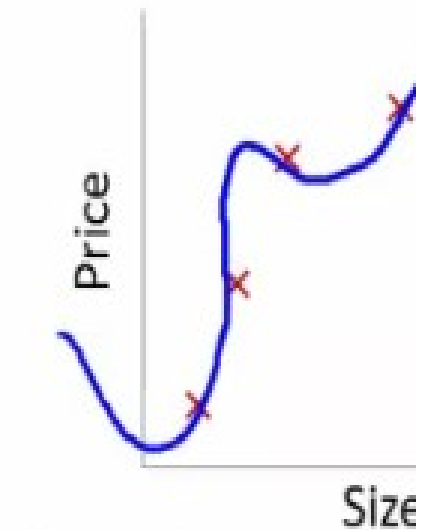
$$\theta_0 + \theta_1 x$$

High bias



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

"Just right"



$$\theta_0 + \theta_1 x + \theta_2 x^3$$

High var

Bias-Variance Tradeoff

- If the model is too simple and has very few parameters then it may have high bias and low variance
- If the model has large number of parameters it may have high variance and low bias
- We need to find a right/good **balance** without overfitting or underfitting the data
- As more parameters are added to a model
 - Complexity of the model rises
 - Variance becomes primary concern while bias falls steadily.

Bias-Variance Tradeoff

- Suppose a training set consists of points x_1, \dots, x_n and real values y_i associated with each point x_i
- We assume there is a function $y = f(x) + \varepsilon$, where the noise ε has zero mean and variance σ^2
- Find $\hat{f}(x)$, that approximates $f(x)$ as well as possible
- To measure how well the approximation was performed, we minimize the mean square error $(y - \hat{f}(x))^2$
- A number of algorithms exist to find $\hat{f}(x)$, that generalizes to points outside of our training set

Bias-Variance Tradeoff

- Variance measures how far a set of (random) numbers are spread out from their average value.
- Measured as expectation of the squared deviation of a random variable from its mean.

$$\text{Var}(X) = E[(x - \mu)^2]$$

$$\text{Var}(X) = E[(x - E[x])^2]$$

$$= E[x^2 - 2xE[x] + E[x]^2]$$

$$= E[x^2] - 2E[x]E[x] + E[x]^2$$

$$= E[x^2] - E[x]^2$$

Bias-Variance Tradeoff

- Turns out expected (mean squared) error of \hat{f} on an unseen sample in general can be decomposed as:

where,

$$E \left[\left(y - \hat{f}(x) \right)^2 \right] = (\text{Bias}[\hat{f}(x)])^2 + \text{Var}[\hat{f}(x)] + \sigma^2$$

$$\begin{aligned} \text{Bias}(\hat{f}(x)) &= E[\hat{f}(x) - f(x)] \\ &= E[\hat{f}(x)] - E[f(x)] = E[\hat{f}(x)] - f(x) \end{aligned}$$

$$\text{Since } f \text{ is deterministic, } E[f] = f$$

and

$$\text{Var}[\hat{f}(x)] = E[\hat{f}(x)^2] - E[\hat{f}(x)]^2$$

Note that all three terms are positive

Bias-Variance Tradeoff

- Notations:

$$\text{Var}[x] = E[x^2] - (E[x])^2$$

$$E[X^2] = \text{Var}(X) + (E[x])^2$$

Given $y = f + \varepsilon$ and $E[\varepsilon] = 0$, $E[y] = E[f + \varepsilon] = E[f] = f$

Since $\text{Var}[\varepsilon] = \sigma^2$, $\text{Var}[y] = E[(y - E[y])^2] = E[(y - f)^2]$

$$= E[(f + \varepsilon - f)^2] = E[\varepsilon^2] = \text{Var}[\varepsilon] + (E[\varepsilon])^2 = \sigma^2$$

Bias-Variance Tradeoff

- The expected error on an unseen sample x can be decomposed as:

$$\begin{aligned}
 E[(y - \hat{f})^2] &= E[(f + \varepsilon - \hat{f})^2] \\
 &= E[(f + \varepsilon - \hat{f} + E[\hat{f}] - E[\hat{f}])^2] \\
 &= E[(f - E[\hat{f}])^2] + E[\varepsilon^2] + E[(E[\hat{f}] - \hat{f})^2] \\
 &\quad + 2E[(f - E[\hat{f}])\varepsilon] + 2E[\varepsilon(E[\hat{f}] - \hat{f})] + 2E[(E[\hat{f}] - \hat{f})(f - E[\hat{f}])] \\
 &= (f - E[\hat{f}])^2 + E(\varepsilon^2) + E[(E[\hat{f}] - \hat{f})^2] \\
 &\quad + 2(f - E[\hat{f}])E(\varepsilon) + 2E(\varepsilon)E(E[\hat{f}] - \hat{f}) + 2E[E[\hat{f}] - \hat{f}](f - E[\hat{f}])
 \end{aligned}$$

Bias-Variance Tradeoff

$$\begin{aligned} &= (f - E[\hat{f}])^2 + E[\varepsilon^2] + E[(E[\hat{f}] - \hat{f})^2] \\ &= (f - E[\hat{f}])^2 + \text{Var}[y] + \text{Var}[\hat{f}] \\ &= \text{Bias}[\hat{f}]^2 + \text{Var}[y] + \text{Var}[\hat{f}] \\ &= \text{Bias}[\hat{f}]^2 + \sigma^2 + \text{Var}[\hat{f}] \end{aligned}$$

- Hence the derivation.

Avoiding Overfitting

- A variety of techniques to avoid overfitting:
 - Cross-validation
 - Regularization
 - Feature selection
 - Dimensionality reduction

Non-exhaustive Cross-validation

Exhaustive Cross-validation

Nested Cross-validation

- Popular way to tune parameters of an algorithm
- One version: k-fold cross validation with validation and test set
- Lets say parameter X needs tuning
 - Possible values 10, 20, 30, 40, 50



Nested Cross-validation

- $k = 7$ in our example
 - One set each picked as Test and Validation, $(k-2)$ picked for training
- For the picked Test set
 - Perform k -fold cross validation on Train & Validation set [Here $k = 6$]
 - Compute the average training error for each value of X
 - Pick the best X
- Repeat for each possible Test set [i.e. 7 times]
- Pick X that was returned maximum times to outer loop

Regularization

Regularization

- Let $\hat{f}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^2 + \theta_4 x^3$
- We want to minimize the MSE:

$$\frac{1}{m} * \min_{\theta_0, \theta_1, \theta_2, \theta_3, \theta_4} \sum_{i=1}^m (\hat{f}_{\theta}(x^{(i)}) - y^{(i)})^2$$

- where m is the number of training samples, theta's are the weight parameters
- Let MSE be represented by $J(\theta)$
- Lets say we want to penalize the higher order terms (2 and 3)

Regularization

- Can add penalty terms say $+1000\theta_3 + 1000\theta_4$
- The effect of this would be that θ_3 and θ_4 need to be quite small to minimize error
- A significantly high penalty can actually convert a overfit problem to an underfit problem
 - Since all the terms with high regularization parameter would become 0 or close to 0
 - E.g. if all terms except θ_0 have a high enough regularization parameter then $\hat{f}(x)$ can become a constant !!!

Feature Selection

- Filter methods: ...
- Wrapper methods: ...
 - Recursive Feature Elimination
- Embedded methods: ...

Dimensionality Reduction

Data Preprocessing

- A popular methodology in data mining is Cross Industry Standard Process for data mining (CRISP DM)
- ...

Feature Engineering

- ...
- **One-hot-encoding or one-of-K:** Refers to splitting the column which contains numerical *categorical data* to many columns depending on the number of categories present in that column.
 - Each column contains “0” or “1” corresponding to which column it has been placed.

Feature Engineering

Fruit	Categorical value of fruit	Price
apple	1	5
mango	2	10
apple	1	15
orange	3	20

- After one hot encoding

apple	mango	orange	price
1	0	0	5
0	1	0	10
1	0	0	15
0	0	1	20

Feature Engineering

- ...