Fundamentals of Machine Learning [DSE 2222]

Department of Data Science and Computer Applications

MIT, Manipal

January 2025

Slide Set 1 – Introduction to course

Introduction to DSE 2222

A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

- Tom Mitchell

Field of study that gives computers the ability to learn without being explicitly programmed.

- Arthur Samuel (1959).

History

- 1950s – Arthur Samuel's checkers playing games
- 1960s
  - Rosenblatt proposed a perceptron
  - Delta Learning Rule
  - Minsky and Papert
- 1970s
  - Symbolic concept of Induction
  - Expert systems
  - Ross Quinlan's ID3
  - Symbolic Natural Language Processing

History

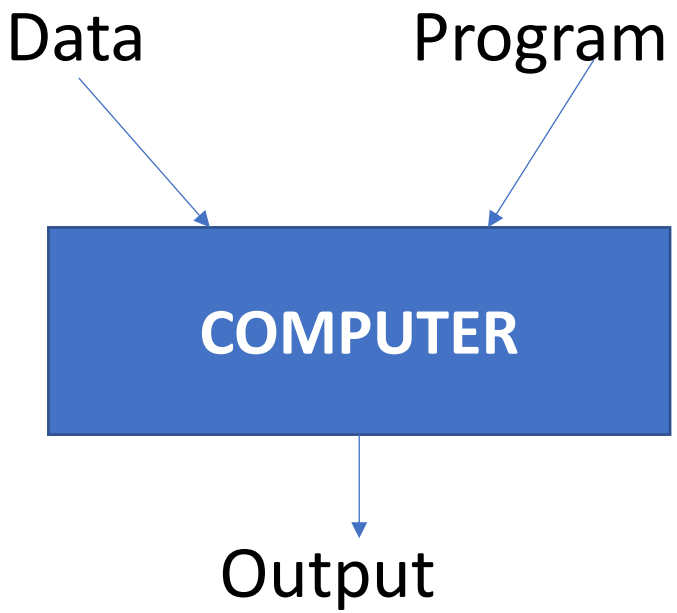
- 1980s
  - Advanced Decision Trees and Rule Learning
  - Resurgence in Neural Networks – Back Propagation
  - Werbos - Multilayer Perceptron Networks
  - Valiant's Probably Approximate Correct Learning (PAC) Learning Theory – experimental methodology

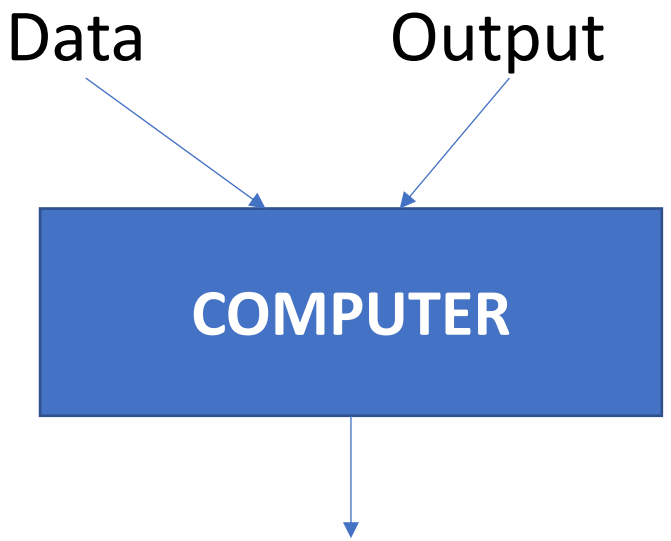
- 1990s
  - Vapnik, Cortes – Support Vector Machines, Kernels in 2000
  - Data Mining
  - Freund & Shapire – ADA Boost – create a strong classifier from an ensemble of weak classifiers
  - Bremen - Random Forest - 2001
  - Bayes Net
  - Adaptive Agents and Web Applications

History

- 2000s
  - Deep Learning
  - New Hardware – GPUS
  - Cloud Enabled
  - Availability of Big Data
  - Milestones
    - 1997 – Deep Blue beat Gary Kasparov in a game of chess
    - 2009 – Google built Self Driving Cars
    - 2011 – IBM’s Watson won the popular game of jeopardy
    - 2014 – Human Vision surpassed by ML vision

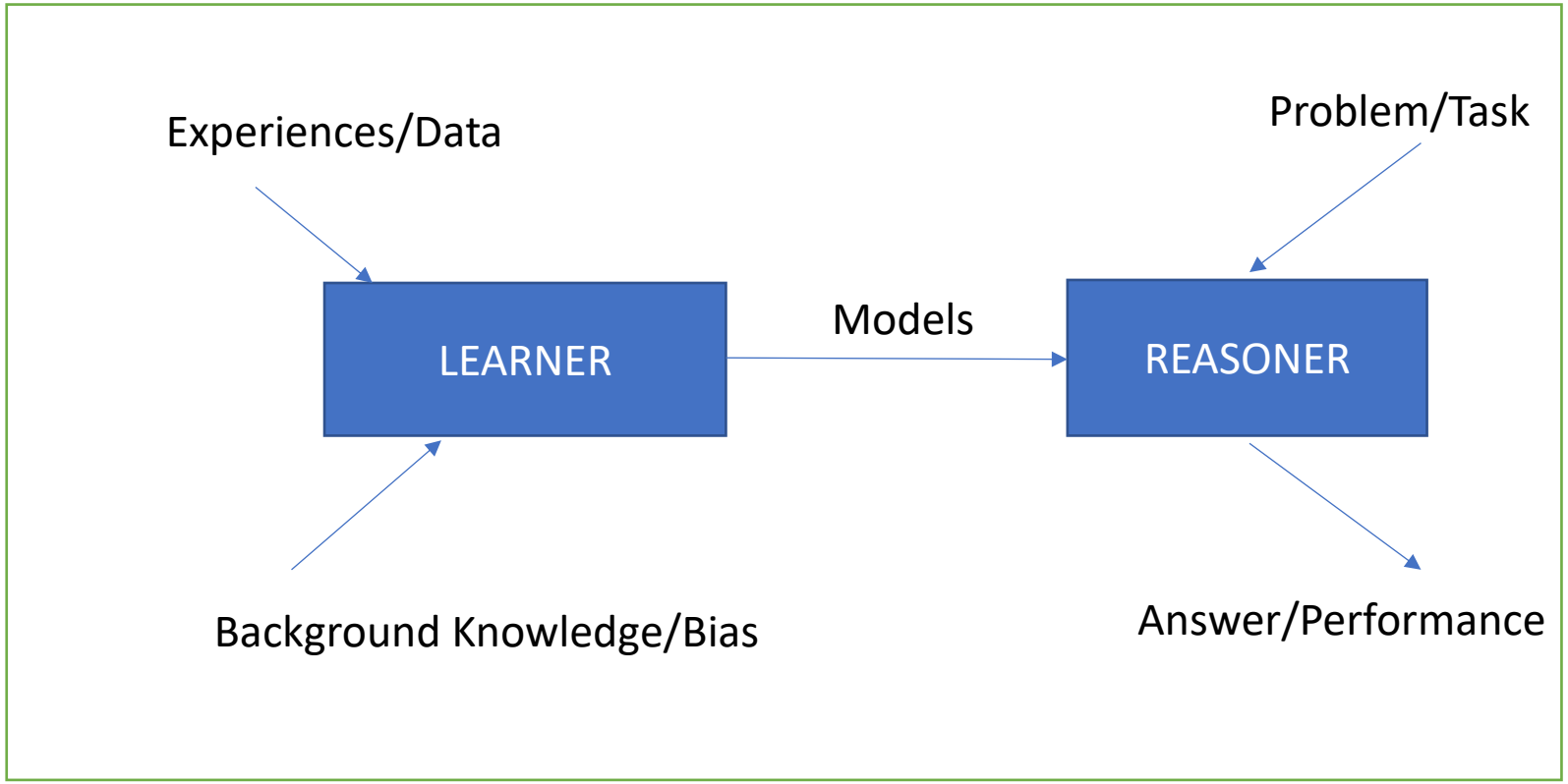
Programmatic Vs. Machine Learning Solution





to solve subsequent tasks

Definition of Machine Learning



Course Outcomes

- Apply machine learning concepts, including supervised, unsupervised, and reinforcement learning, to solve regression and classification problems by evaluating model performance using accuracy and error metrics.
- Analyze supervised learning algorithms like logistic regression, k-nearest neighbors, and Naïve Bayes for classification and regression, considering bias-variance trade-offs and cost functions.
- Evaluate advanced supervised learning techniques, including decision trees, support vector machines, and artificial neural networks, with a focus on their implementation, optimization, and applicability.
- Apply unsupervised learning methods such as clustering and dimensionality reduction, along with ensemble techniques like boosting, bagging, and stacking, to address challenges like anomaly detection and class imbalance.

DSE 2222 FUNDAMENTALS OF MACHINE LEARNING [3 0 0 3]

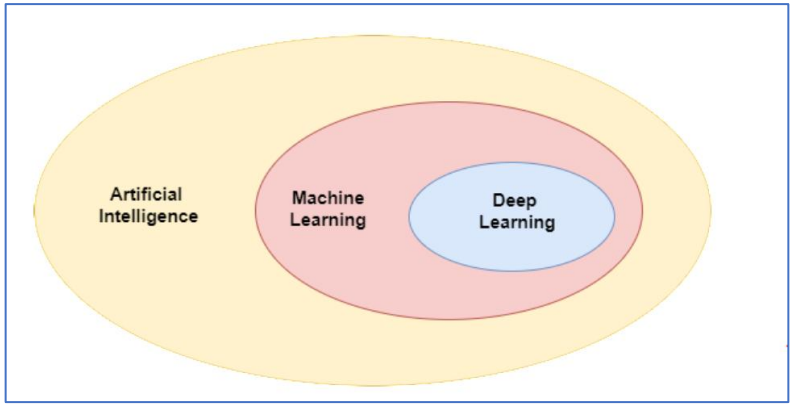
Machine Learning Basics: Types of Machine Learning, Supervised vs. Unsupervised Learning, Parametric vs. non-parametric models., Instance Based learning – k-nearest neighbors, Simple Regression Models: Linear, Logistic, Cost functions, Gradient Descent, Batch Gradient Descent, Overfitting, Model Selection, No free lunch theorem, bias/variance trade-off, union and Chernoff bounds, VC dimensions. Bayesian Models: Bayesian concept learning, Bayesian Decision Theory, Naïve Bayesian, Laplacian Correction, Bayesian Belief Networks. Tree Models: information theory, decision tree induction, tuning tree size, ID3,C4.5, CHAID, Decision Stump. Support Vector Machines: kernel functions, Regression Models: Ridge and Lasso Regression, GLM and theexponential Family. Bagging algorithm, Random Forests, Grid search and randomized grid search, Partial dependence plots. Ensembling and Boosting Algorithms: Concept of weak learners, Adaptive Boosting, Extreme Gradient Boosting (XGBoost). Artificial Neural Networks: Perceptron, Back propagation, Hopfield Network. Curse of Dimensionality: Factor Analysis, Principal Component Analysis (PCA), Difference between PCAs and Latent Factors

References

- K. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
- G. James, D. Witten, T Hastie, R Tibshirani, *An introduction to statistical learning with applications in R*, Springer, 2013.
- J. Han, M. Kamber, J. Pei, *Data Mining concepts and techniques*, (2e), Morgan Kaufmann-Elsevier, 2011.
- T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, (2e), Springer, 2009.
- T. M. Mitchell, *Machine Learning*, (Indian Edition), MacGraw Hill, 2017.
- C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 2019

INTRODUCTION TO ANALYTICS AND MACHINE LEARNING

- Analytics is a collection of techniques and tools used for creating value from data.
- Techniques include
  - Artificial intelligence (AI): Algorithms and systems that exhibit human-like intelligence.
  - Machine learning (ML): Subset of AI that can learn to perform a task with extracted data and/or models.
  - Deep learning (DL): Subset of machine learning that imitate the functioning of human brain to solve problems.



What is Machine Learning?

- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

Classification of machine learning algorithms

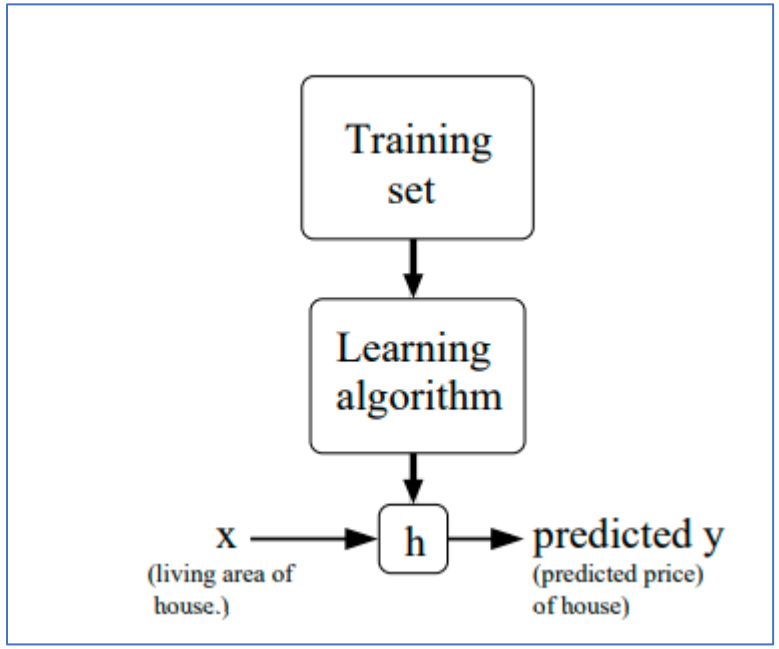
- Association
- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
- Reinforcement Learning

Learning Associations

- Basket analysis:
  - P (Y | X) probability that somebody who buys X also buys Y where X and Y are products/services.

Supervised learning

- **Supervised learning**
  - predict y from x
  - Given a labelled set of input-output pairs, Map input x to output y
  - Given: Training set $\{(x_i , y_i) \mid i = 1 \ldots n\}$
  - Find: A good approximation to
    f : X → Y where $y \in \{1, \ldots, C\}$,
  - Classification – y is categorical
  - Regression – y is real values
  - Formalize the problem is as function approximation.
    - y = f(x) for some unknown function f
    - goal of learning is to estimate the function f
    - given a labeled training set
    - then to make predictions using ˆy = ˆ f(x).
  - Generalization-to make predictions on novel inputs
  - Examples - spam detection, Digit Recognition, stock prices



Example

(a) (b) (c)

Figure 1.3 Three types of iris flowers: setosa, versicolor and virginica. Source: `http://www.statlab.uni-heidelberg.de/data/iris/` . Used with kind permission of Dennis Kramb and SIGNA.

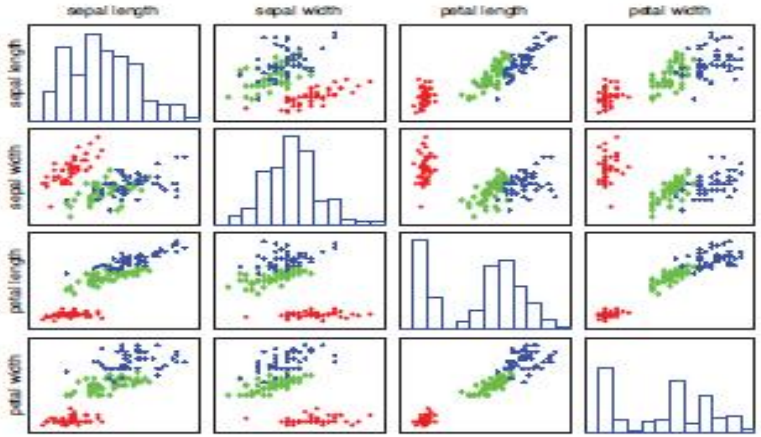
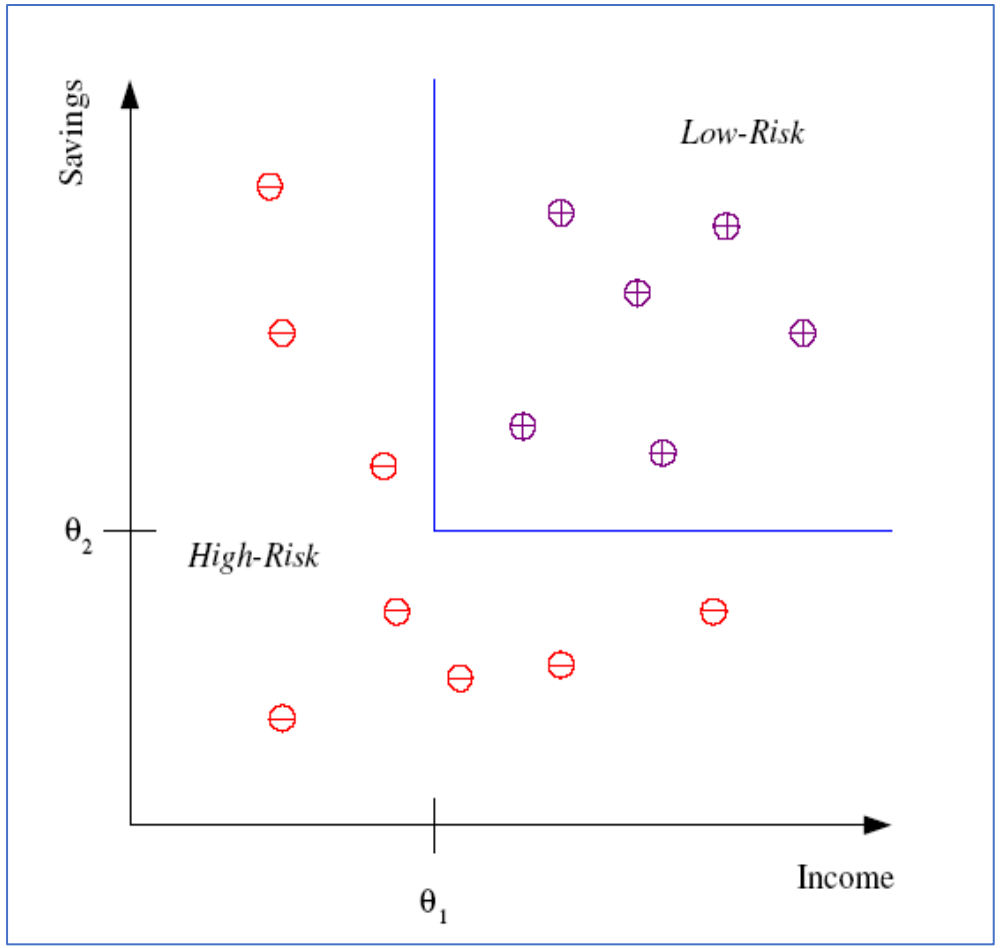


Figure 1.4 Visualization of the Iris data as a pairwise scatter plot. The diagonal plots the marginal histograms of the 4 features. The off diagonals contain scatterplots of all possible pairs of features. Red circle = setosa, green diamond = versicolor, blue star = virginica. Figure generated by `fisheririsDemo`.

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their income and savings



Discriminant: IF *income* $> \theta_1$ AND *savings* $> \theta_2$
THEN low-risk ELSE high-risk

Classification: Applications

- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
- Use of a dictionary or the syntax of the language.
- Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- ...

Training examples of a person

Test images

Regression

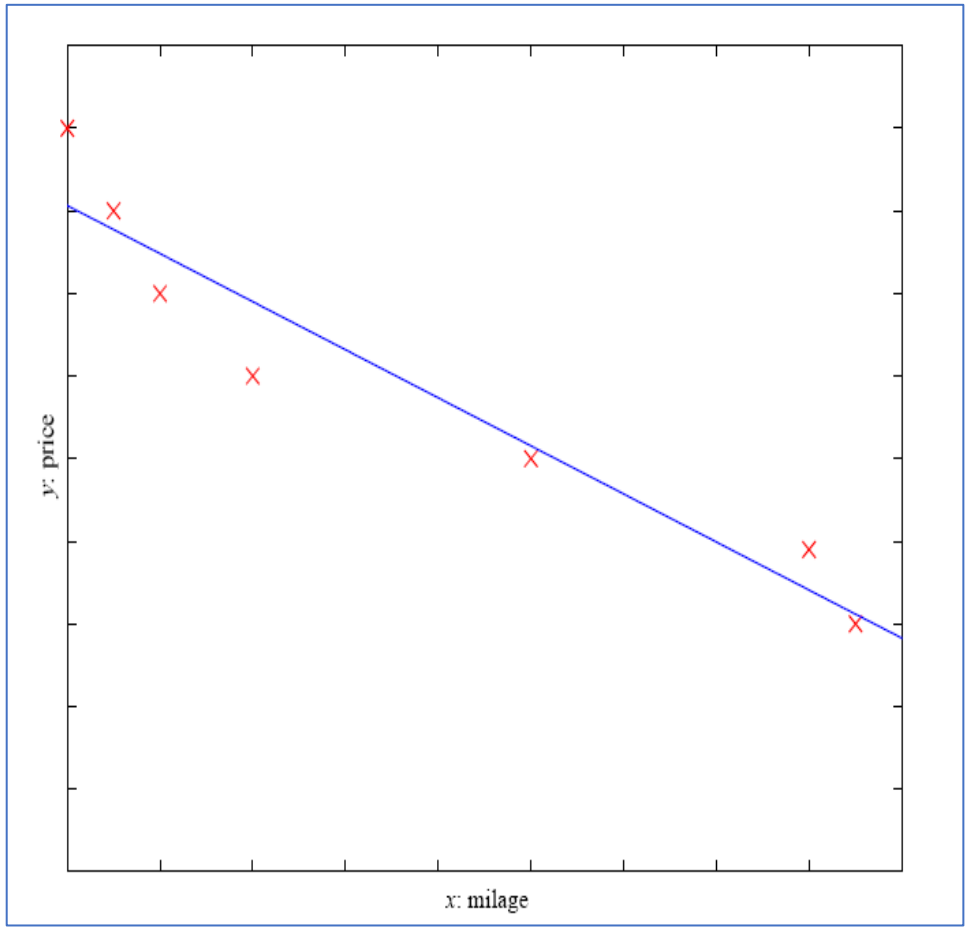
- Example: Price of a used car
- x : car attributes

  y : price

  $y = g(x \mid \theta)$

  g ( ) model,

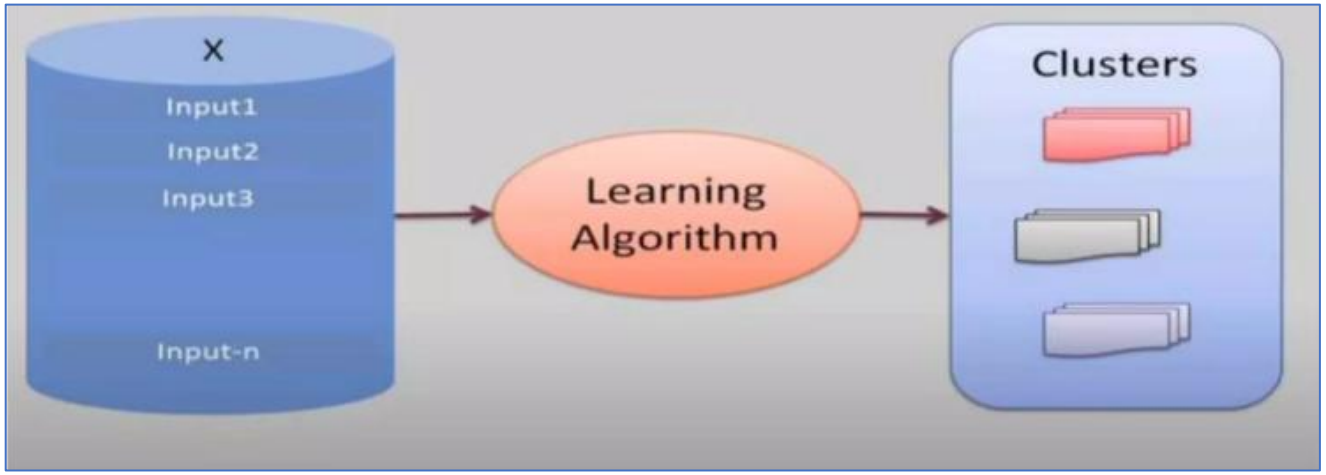
  θ parameters



Supervised Learning: Uses

- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: The rule is easy to understand
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Unsupervised learning



- Model p(x), evaluate how likely x is, understand x
- Find interesting patterns or knowledge discovery
- Clustering, patterns

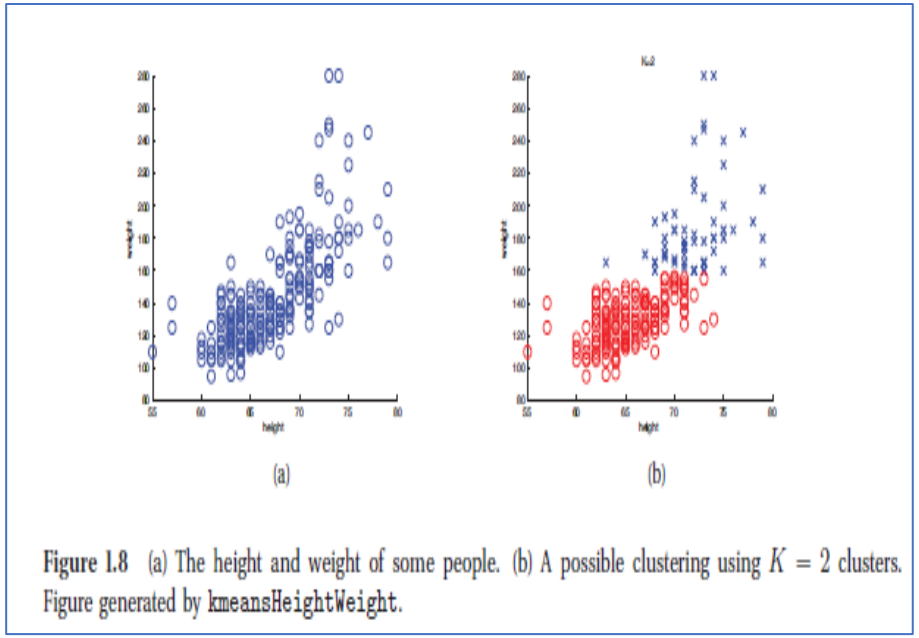
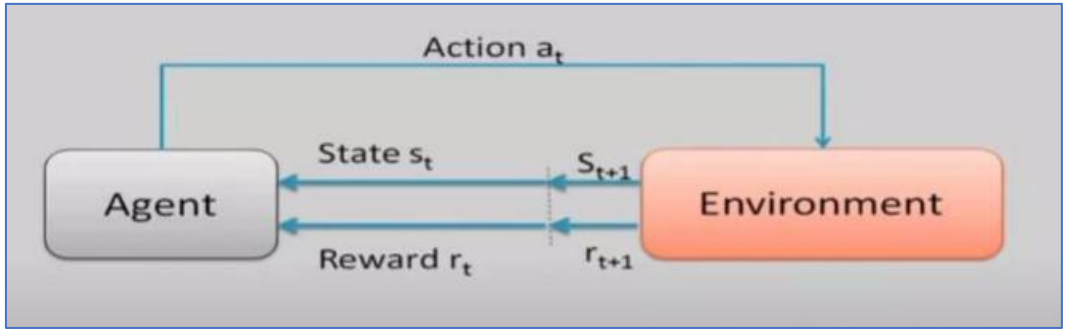


Figure 1.8 (a) The height and weight of some people. (b) A possible clustering using $K = 2$ clusters. Figure generated by `kmeansHeightWeight`.

Reinforcement Learning

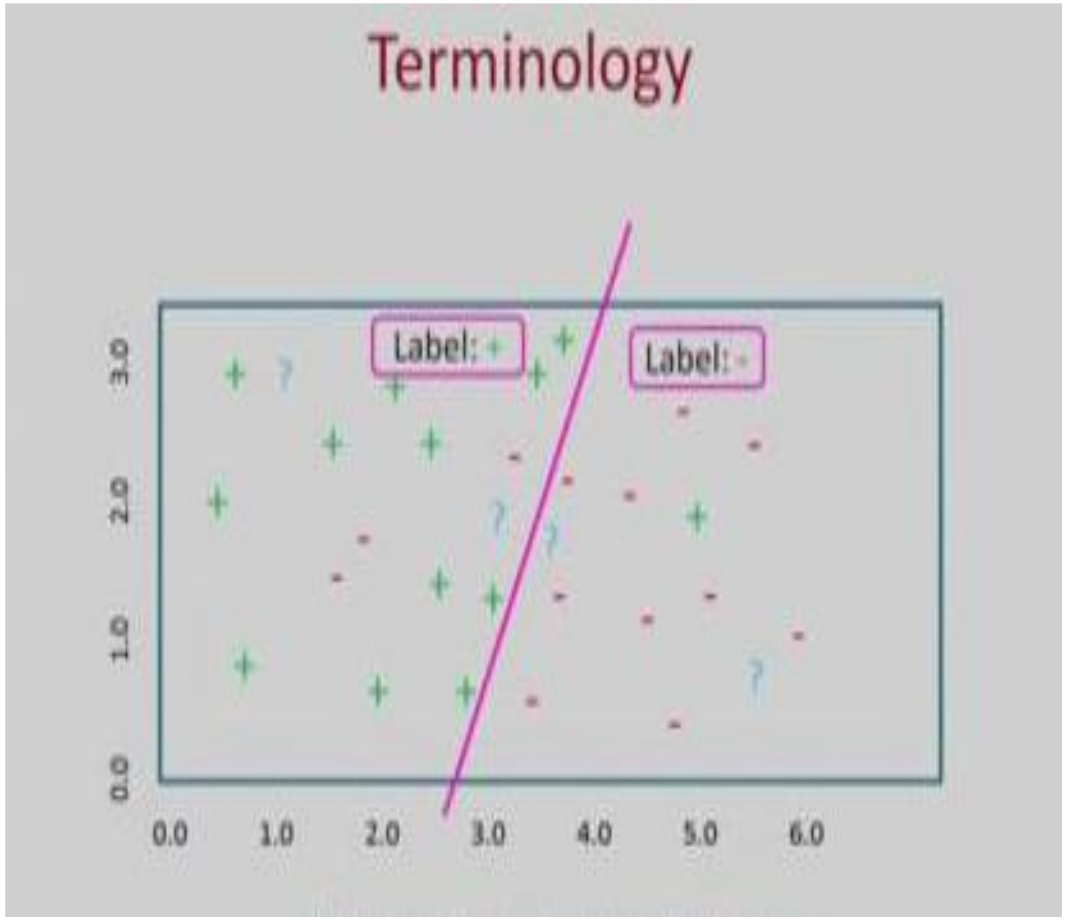
- Learning a policy: A sequence of outputs
- No supervised output but delayed reward
- Credit assignment problem
- Game playing
- Robot in a maze
- Multiple agents, partial observability, ...

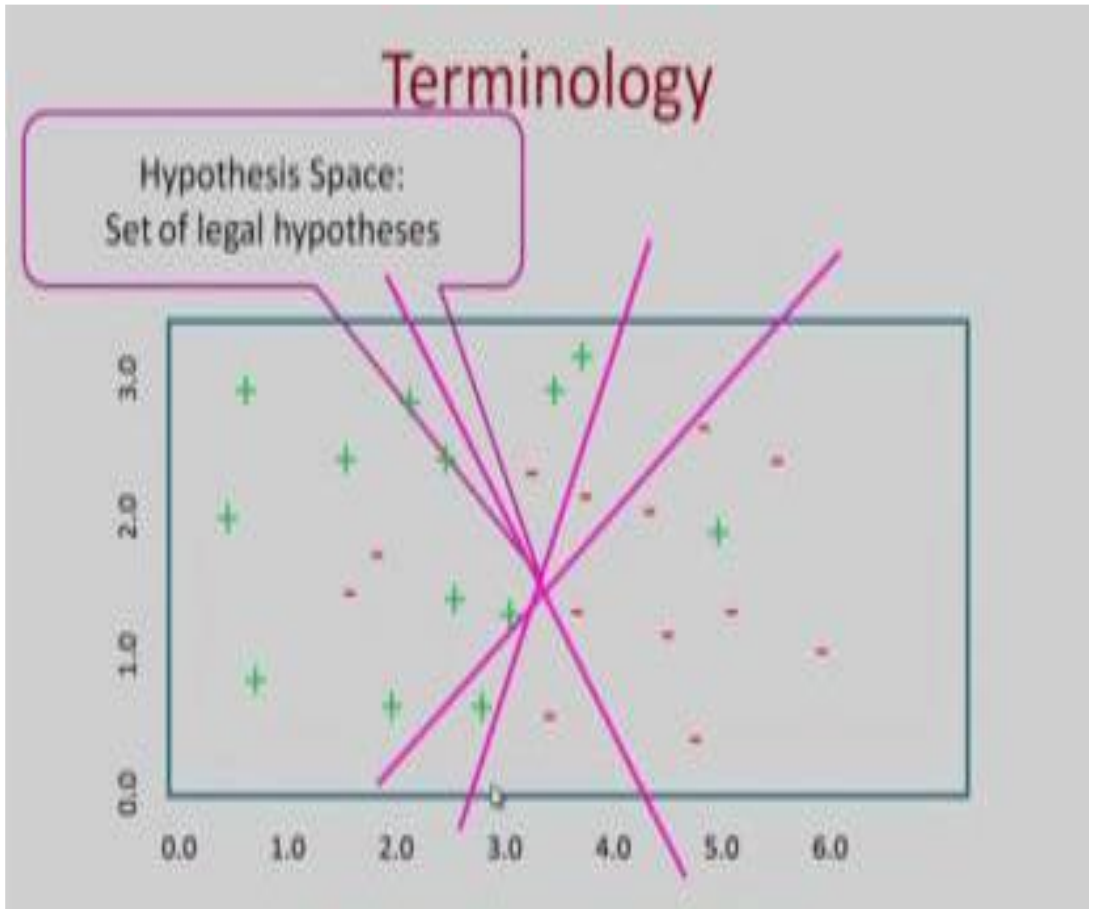


Terminologies

- Feature Space represents the data in a set of n features
- Features are properties that describe each instance in Instance space
- Multiple features represented in a Feature Vector
- We are given data and induction identifies a function, which can explain the data.
- Hypothesis could be a function which is a line or curve which separates classes.
- Hypothesis Space
  - is the set of all legal functions that are solutions to the task
  - Comprises of the features chosen and the language or the class of functions

Terminology



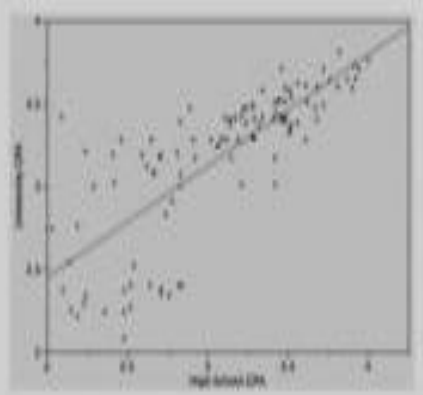


Hypothesis language - representations

Representations

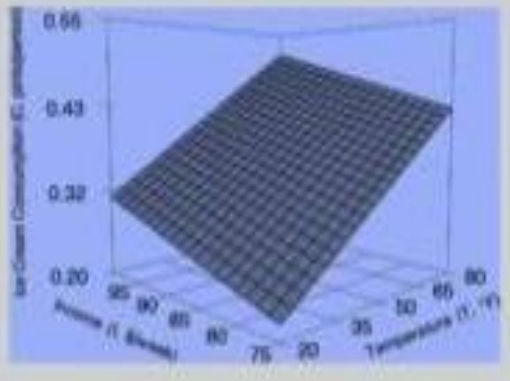
1. Decision Tree

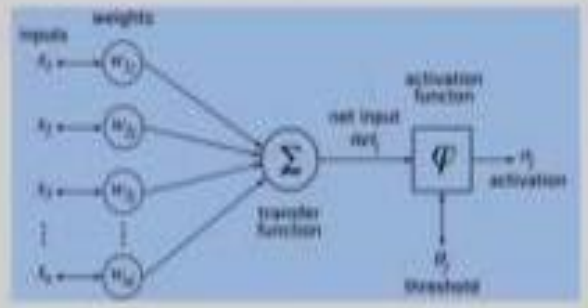
2. Linear function

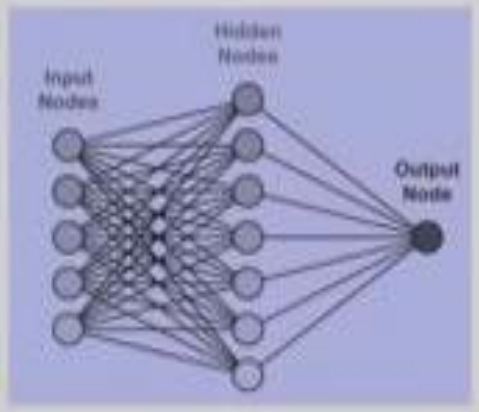


Representations

3. Multivariate linear function
4. Single layer perceptron
5. Multi-layer neural networks







Inductive Bias

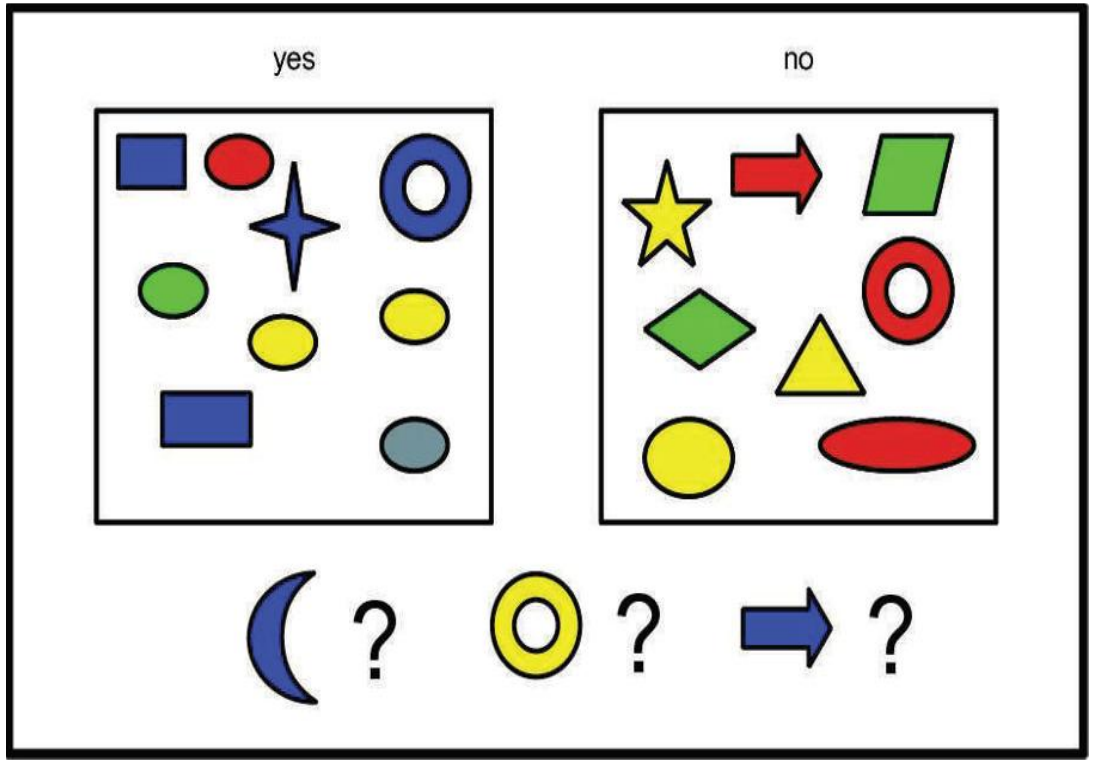
- Need to make assumptions
  - Experience alone does not allow us to make conclusions about unseen data instances
- Examples
  - Occam's Razor
    - prefer the simplest hypothesis.
    - Philosophical principle that if something can be described in a short language that hypothesis is to be preferred over a more complex hypothesis.
  - minimum description length
  - Maximum margin
- Two types of Bias
  - Restriction :
    - Limit the Hypothesis space
    - For instance, Hypothesis language reflects the inductive bias of the learner
  - Preference
    - Impose ordering on Hypothesis Space

Inductive Learning

- Inducing a general function from training examples.
- Given some training examples, the objective is to generalize to test data
  - Construct hypothesis h to agree with c on the training examples.
  - Consistent Hypothesis agrees with all training examples.
  - hypothesis that is consistence with all the training examples
  - A hypothesis is said to generalize well if it correctly predicts the value of y for novel example.
- Inductive Learning is an ill posed problem
  - where the data by itself is not sufficient to find a unique solutions
- inductive learning hypothesis
  - A hypothesis which has a low training error over a sufficiently large training set is expected to do well on unseen examples.

Formal Definition of Learning

- **function approximation**.
  - We assume $y = f(\mathbf{x})$ for
  - some unknown function f
  - The goal of learning is to estimate the function f
    - given a labeled training set
    - then to make predictions using $\hat{y} = f(x)$.
- Generalization
  - to make predictions on novel inputs, meaning ones that we have not seen before



Leslie Kielbling

Need for Probabilistic Prediction

- the probability distribution over possible labels
- given the input vector **x** and training set *D* is $p(y|\mathbf{x},D)$
- If there are just two classes
  - it is sufficient to return the single number $p(y = 1|\mathbf{x},D)$
  - since $p(y = 1|\mathbf{x},D) + p(y =0|\mathbf{x},D) = 1$.)
- MAP Estimate (Maximum A Posteriori estimate)
  - Most probable class

$$\hat{y} = \hat{f}(\mathbf{x}) = \underset{c=1}{\overset{C}{\operatorname{argmax}}}\, p(y = c|\mathbf{x}, \mathcal{D})$$

Experimental Evaluation of Learning Algorithms

- Evaluating the performance of learning systems is important because
  - Learning systems are designed to predict the class of future unlabeled data points

I. Experimental Model Evaluation include

- Error
- Accuracy
- Precision/Recall

II. Typical Choices for Sampling Methods

- Train/test sets
  - Error or other parameters is measured on the training set
  - the error or accuracy on the training set is not, may not be a reflection of the true error
- K-fold Cross-validation

Experimental Model Evaluation for Prediction

- How is error measured?
- Suppose we want to make a prediction of a value for a target feature on example x
  - y is the observed value of target feature on example x
  - $\hat{y}$ is the predicted value of targer feature on example x
  - $\hat{y} = h(x)$
  - For Regression Problems
  - Absolute error = $\frac{1}{n}\sum_{i=1}^{n}|h(x) - y|$
  - Sum of square error = $\frac{1}{n}\sum_{i=1}^{n}(h(x) - y)^2$

Experimental Model Evaluation for classification

- Suppose we want to make a prediction of a value for a target feature on example x
  - y is the observed value of target feature on example x
  - $\hat{y}$ is the predicted value of targer feature on example x
  - $\hat{y} = h(x)$
- For Classification Problems
- Number of Misclassifications:
- $\frac{1}{n}\sum_{i=1}^{n} \partial(h(x), y)$
  - returns 1 if the class labels are different
  - Returns 0 if the class labels are same

Model Evaluation in Classification

True Class ---------- →

Hypothesis Class- →

TP	FP
FN	TN
P	**N**

- Accuracy = $\frac{TP+TN}{P+N}$
- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{P}$

Ex 1 - Compare models

Model A

Predicted \ Actual	1	0	Totals
1	79	28	107
0	72	213	285
Totals	151	241	392

Model B

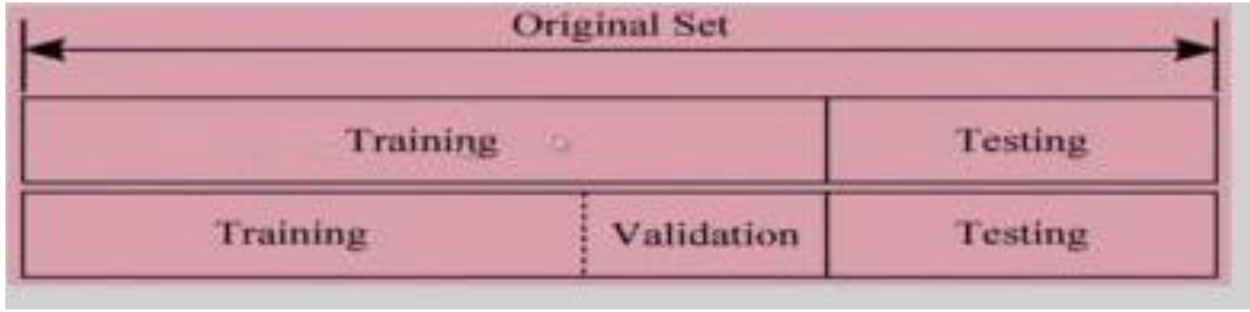
Predicted \ Actual	1	0	Totals
1	140	38	178
0	11	203	214
Totals	151	241	392

Model C

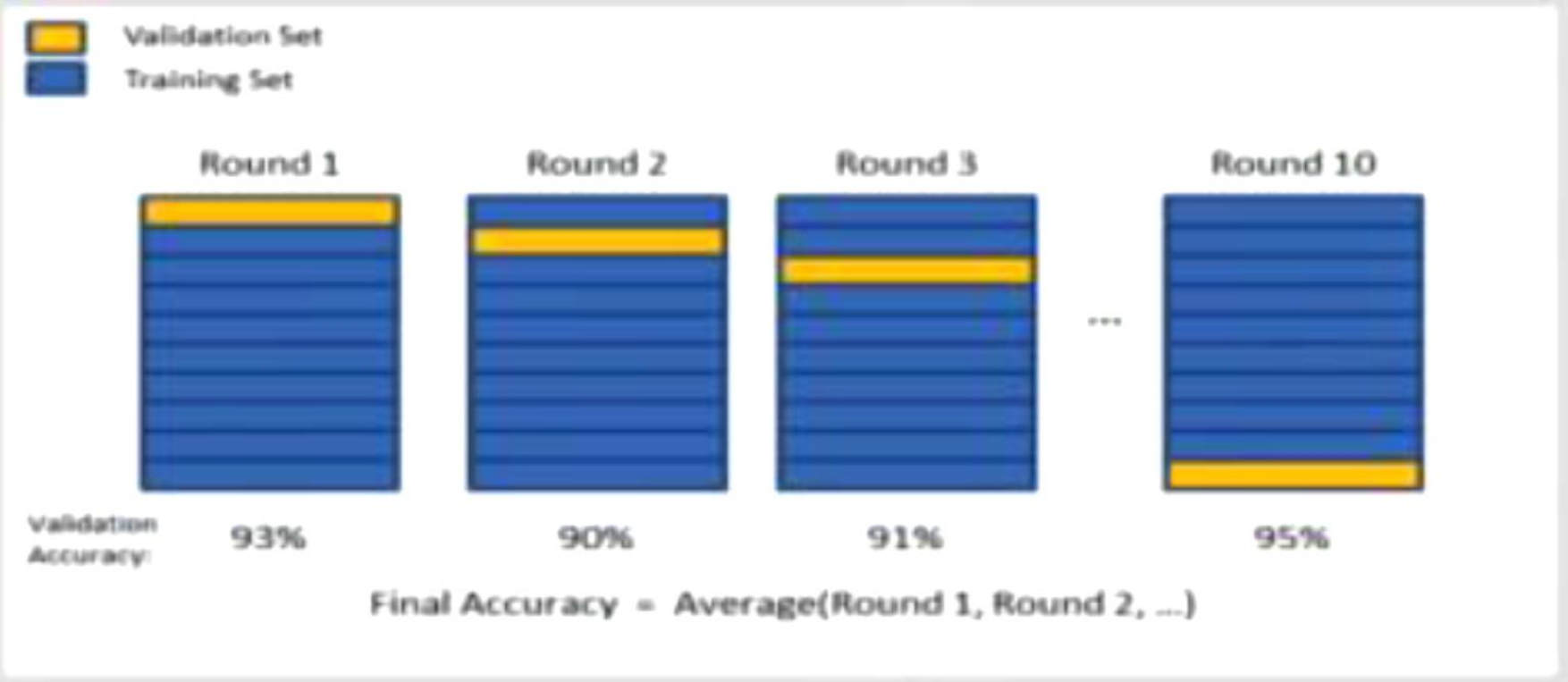
Predicted \ Actual	1	0	Totals
1	129	18	147
0	22	223	245
Totals	151	241	392

Typical Choices for Sampling Methods

K fold cross validation

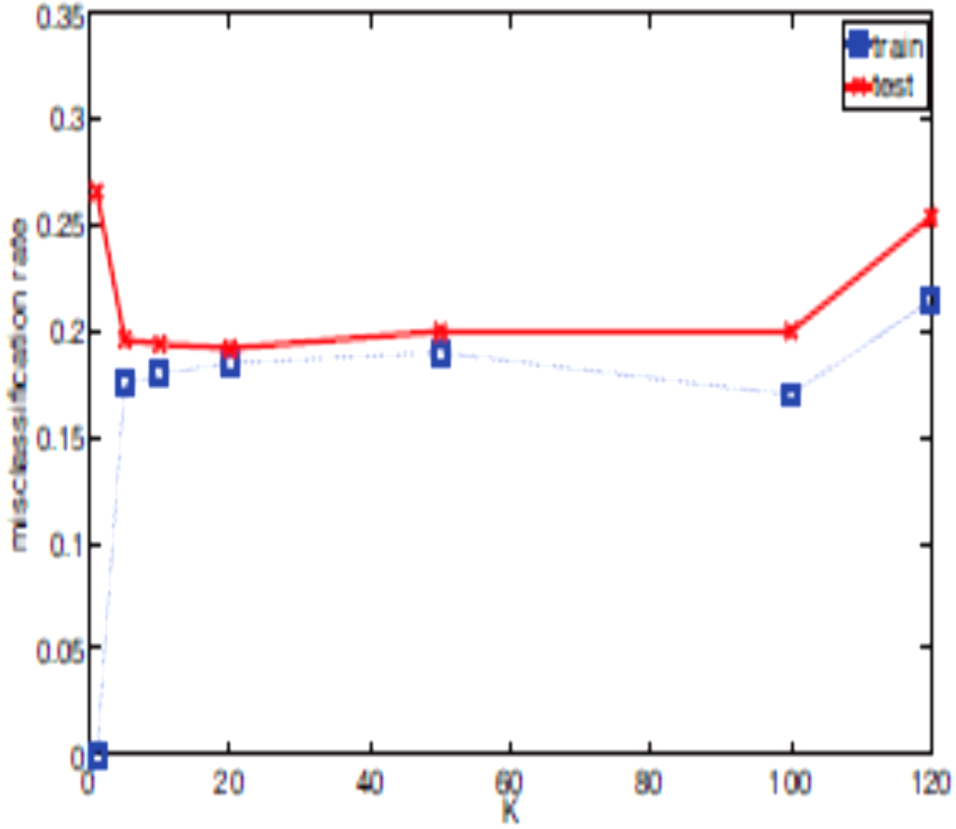


- Validation set used to tune model parameters



Model Selection in the KNN case

- We care about generalization error or test error
- Test error (Red Line)
  - U-shaped curve
  - for complex models (small *K*), the method overfits
  - simple models (big *K*), the method underfits
- Use validation test to pick *K*
  - *Pick* the value with the minimum error on the test set
  - K = 10 to 100



Sample Error vs True Error

- **The Sample Error**
  - of hypothesis f with respect to target function c and data sample S is
  - $\text{error}_S(f) = \frac{1}{n}\sum_{x \in S} \partial(f(x), c(x))$
- **The True Error**
  - of hypothesis f with respect to target function c and distribution D
  - Is the probability that h will misclassify an instance drawn on random according to D
  - $\text{error}_D(f) = \Pr_{x \in D}[f(x) \neq c(x)]$
- **Causes of Error**
  - **Search Bias**
    - given the hypotheses space the search algorithm is not exhaustively searching the hypotheses space, but making certain simplification
  - **Variance**
    - is the amount that the estimate of the target function will change given different training data
  - **Noise**
    - the features used are not sufficient to capture everything about the task

Bias & Variance

- **Bias**
  - is **the simplifying assumptions made by the model to make** the target function easier to approximate.
- Variance
  - is the amount that the estimate of the target function will change given different training data.
  - If the test set is small there will be variance.
- Trade-off is tension between the error introduced by the bias and the variance.

Machine Learning Trade-off

- In ML there is always are trade-off between
  - Complex hypothesis that fit the training data well
  - Simpler hypothesis that may generalize better
- As the amount of training data increases, the generalization error decreases
- No Free Lunch Algorithm (Wolpert 1996).
  - All models are wrong, but some models are useful. — George Box (Box and Draper 1987
- As a consequence of the no free lunch theorem
  - Need to develop many different types of models
  - to cover the wide variety of data that occurs in the real world.
  - For each model, there may be many different algorithms we can use to train the model
  - Then make speed-accuracy-complexity tradeoffs.

Developing a learning algorithm

- To improve the learning algorithm

Get more training examples
Try smaller sets of features
Try getting additional features
Try adding polynomial features
Try decreasing λ
Try increasing λ

Evaluating your hypothesis

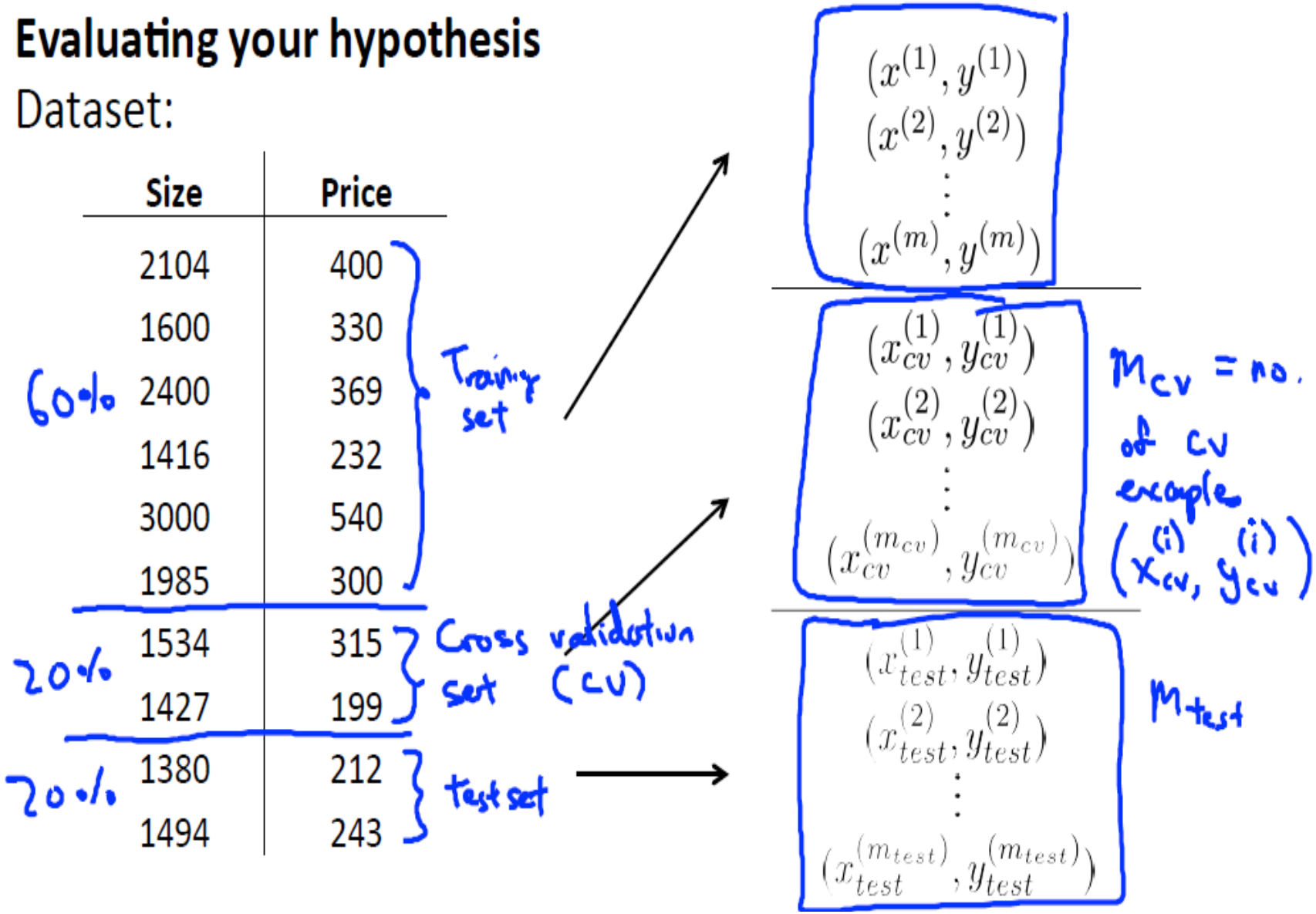
Dataset:

Size	Price	
2104	400	Training set
1600	330	
2400	369	
1416	232	
3000	540	
1985	300	
1534	315	Cross validation set (CV)
1427	199	
1380	212	test set
1494	243	

60%

20%

20%



Train/validation/test error

Training error:

$$J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Cross Validation error:

$$J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

Test error:

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

Model selection

1. $h_\theta(x) = \theta_0 + \theta_1 x$ → $\min_\theta J(\theta)$ → $\theta^{(1)}$ → $J_{cv}(\theta^{(1)})$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ → $\theta^{(2)}$ → $J_{cv}(\theta^{(2)})$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$ → $\theta^{(3)}$ → $J_{cv}(\theta^{(3)})$

⋮

10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$ → $\theta^{(10)}$ → $J_{cv}(\theta^{(10)})$

$d = 4$

Pick $\theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4$

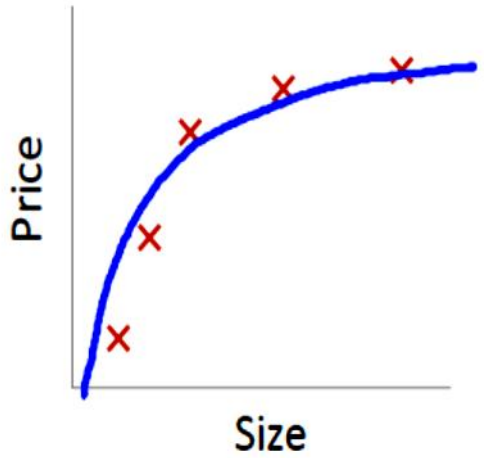
Estimate generalization error for test set $J_{test}(\theta^{(4)})$

Bias vs. Variance

$\theta_0 + \theta_1 x$

High bias
(underfit)

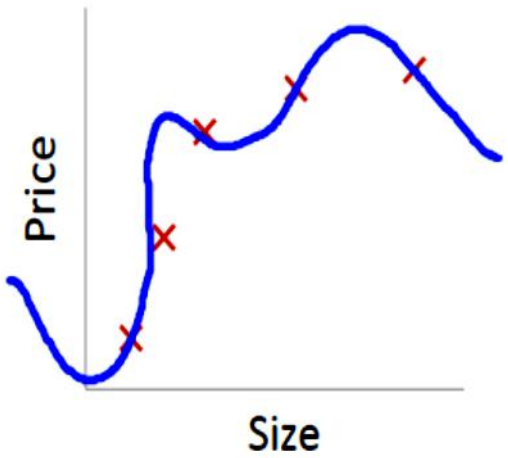
$d = 1$



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

"Just right"

$d = 2$



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

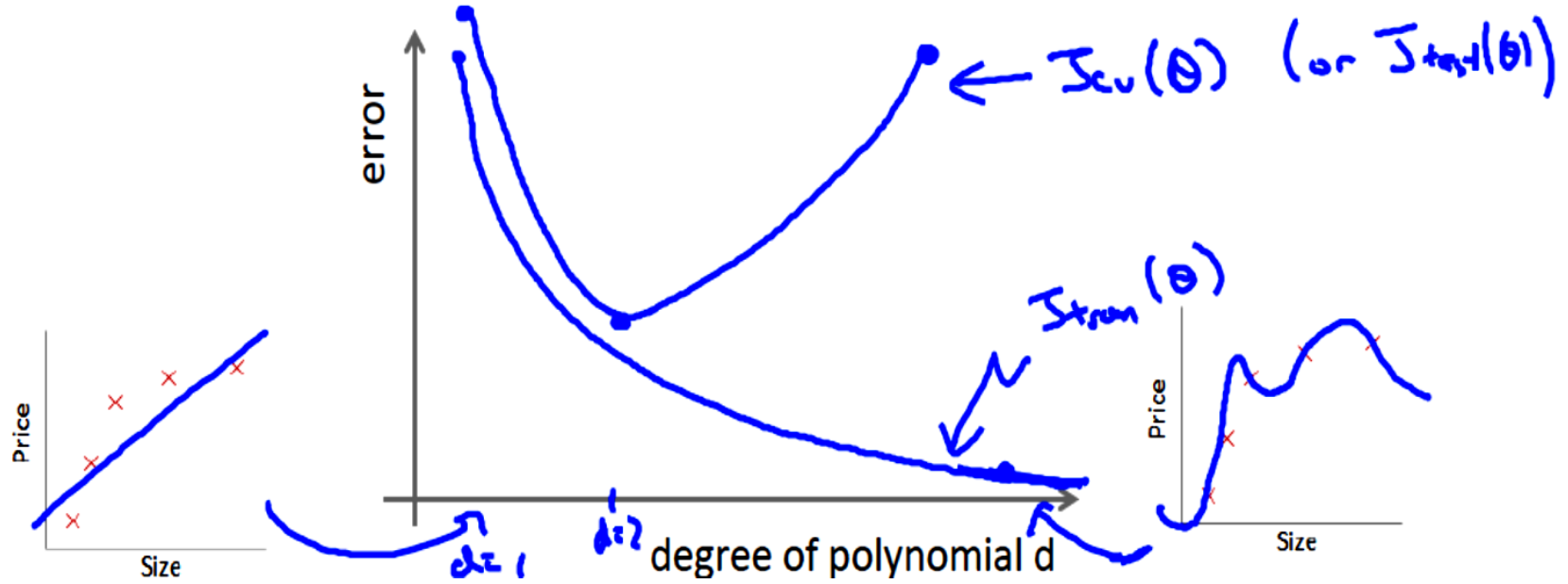
High variance
(overfit)

$d = 4$

Bias Vs Variance

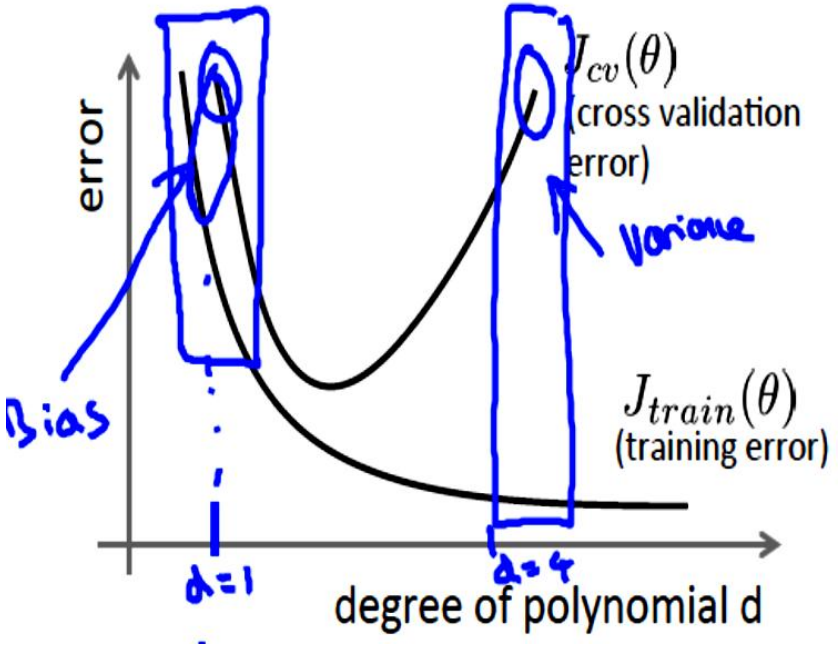
Training error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

Cross validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$ (or $J_{test}(\theta)$)



Diagnosing Bias vs Variance

Suppose your learning algorithm is performing less well than you were hoping. ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high.) Is it a bias problem or a variance problem?



Bias (underfit):

→ $J_{train}(\theta)$ will be high

$J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (overfit):

→ $J_{train}(\theta)$ will be low

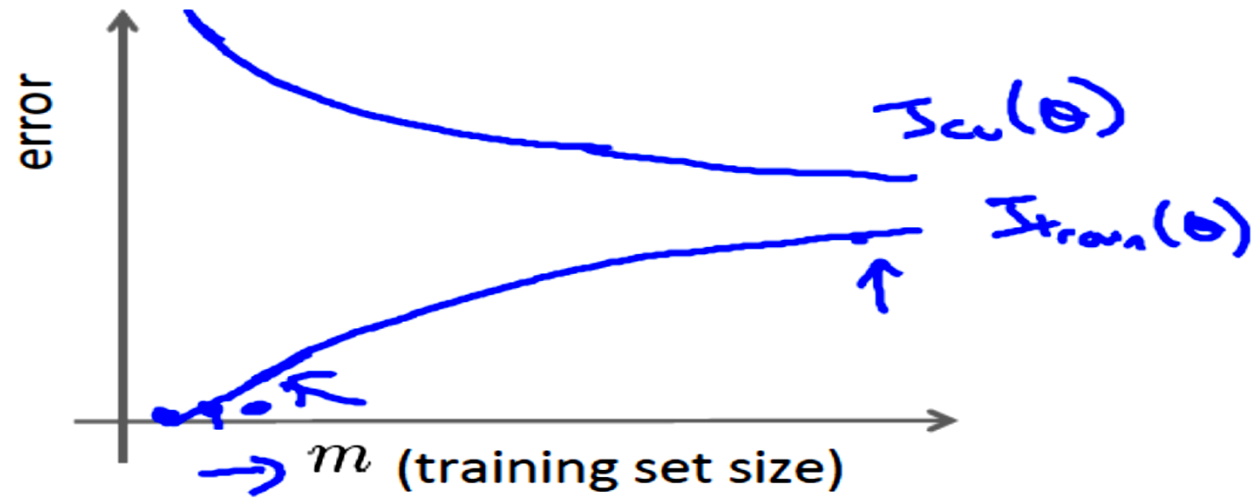
$J_{cv}(\theta) \gg J_{train}(\theta)$

Learning Curves

Learning curves

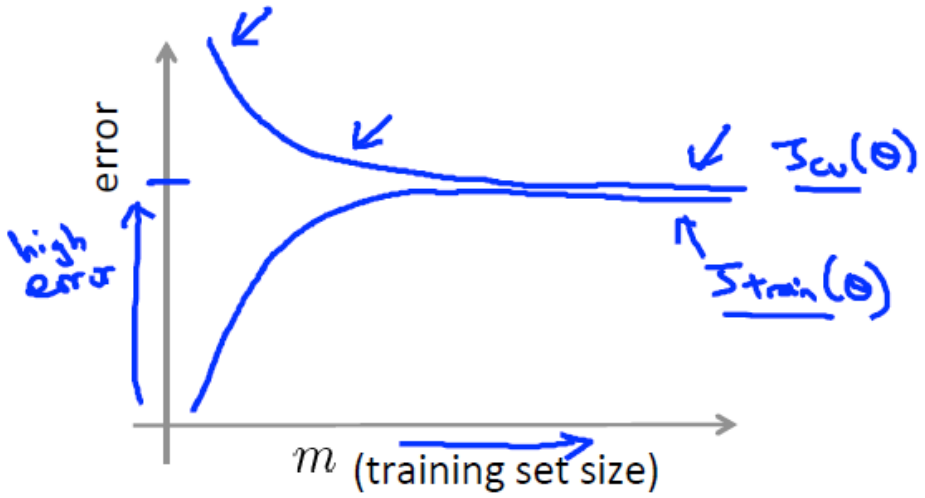
$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$



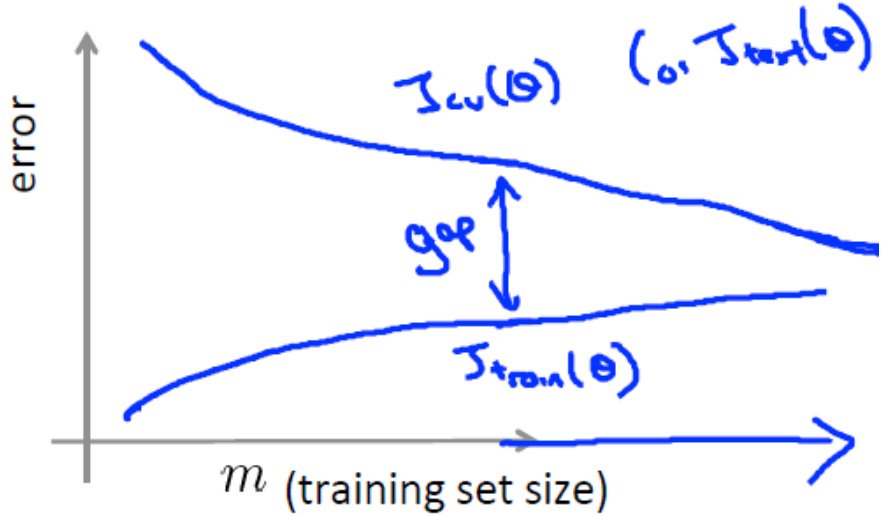
High Bias vs. High Variance

High bias



If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

High variance



If a learning algorithm is suffering from high variance, getting more training data is likely to help.

Solutions

- Get more training examples → fixes high variance
- Try smaller sets of features → fixes high variance
- Try getting additional features → fixes high bias
- Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, \text{etc})$ → fixes high bias.
- Try decreasing λ → fixes high bias
- Try increasing λ → fixes high variance

Parametric vs Non Parametric Models

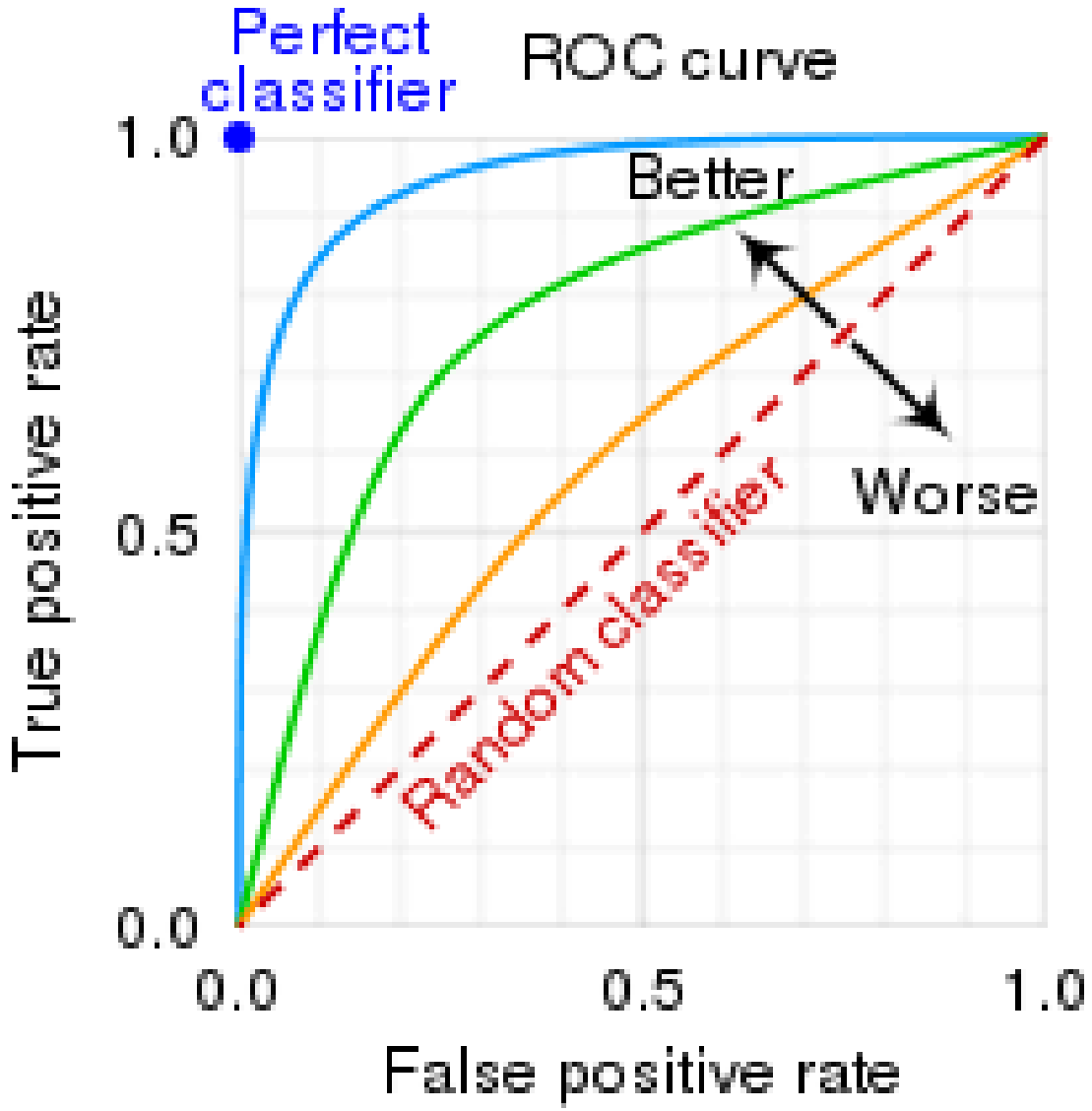
- **Parametric model**
  - A learning model that summarizes data with a set of parameters of fixed size
  - independent of the number of training examples.
  - Faster to use
  - but making stronger assumptions
  - Examples
    - Linear Regression
    - Logistic Regression
    - Naive Bayes

Parametric vs Non Parametric Models

- Nonparametric model
  - Model with lot of data and no prior knowledge
  - no need to choose the right features.
  - more flexible, but often computationally intractable
  - Examples
    - Decision Trees
    - K-Nearest Neighbor
    - Support Vector Machines

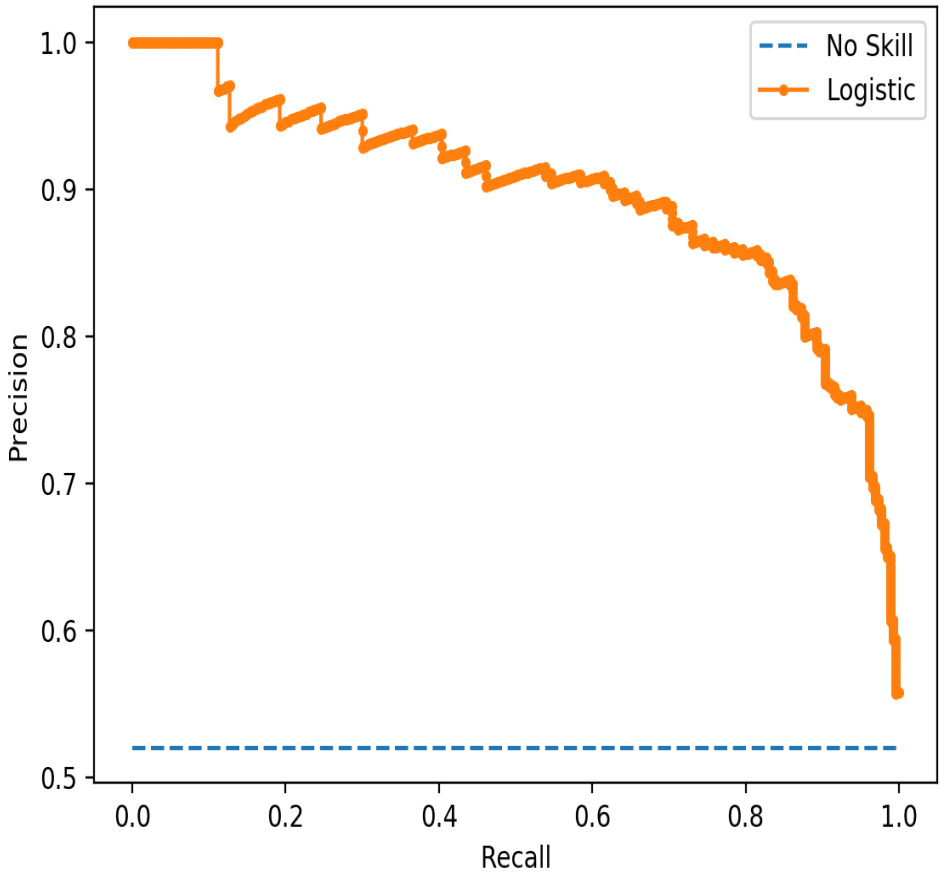
ROC

- ROC
  - is plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0.
  - False Positive Rate = 1 – Specificity
  - The area under the curve (AUC) can be used as a summary of the model skill.
    - A no-skill classifier is one that cannot discriminate between the classes
      - would predict a random class or a constant class in all cases.
      - has an AUC of 0.5.
    - A model with perfect skill is represented at a point (0,1).
      - Has AUC close to 1



PRC

- is a plot of the precision (y-axis) and the recall (x-axis) for different thresholds
- Baseline is P/(P+N)
- Precision
  - is a ratio of the number of TPs divided by the sum of TPs and FPs.
  - describes how good a model is at predicting the positive class.
- Recall is Sensitivity
- composite scores that attempt to summarize the precision and recall are:
  - F-Measure or F1 score- harmonic mean of the precision and recall
  - AUC (PRC)
    - summarizes the integral or an approximation of the area under the precision-recall curve.



Class Imbalance Problem

- Rare classes have less than 10% of instances.
- Most health datasets contain majority negative class and minority positive class
- Target class has much lower precision and recall than majority class
- In general , the algorithms preferred are:
  - instance based learning
  - SVM where decision boundaries are not affected by number of instances
  - MLP
  - Ensembling to aggregate predictions

Class Imbalance Problem

- Preprocessing techniques
  - Under sampling
  - Over sampling
  - Synthetic Minority Over Sampling Technique (SMOTE)
- Cost Benefit Analysis
  - The Cost matrix is used to represent the imbalance
  - For instance, the cost of FNs can be 10 times more than the cost of the FP
  - Cost function to be minimized, as well as reduce the number of FNs or high recall is desirable

Ensemble Learning

general meta approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.

Ensembles overcome three problems Dietterich(2002) –

- **Statistical Problem**
  - hypothesis space is too large for the amount of available data.
  - many hypotheses with the same accuracy on the data
  - risk that the accuracy of the chosen hypothesis is low on unseen data!
- **Computational Problem**
  - the learning algorithm cannot guarantee finding the best hypothesis.
- **Representational Problem**
  - the hypothesis space does not contain any good approximation of the target class(es).

Simple Ensembling Techniques

- Max Voting
  - Mostly used for classification
  - multiple models are used to make predictions for each data point.
  - The predictions by each model are considered as a ‘vote’.
  - The predictions which we get from the majority of the models are used as the final prediction.
- Averaging
  - can be used for making predictions in regression or while calculating probabilities for classification
  - multiple predictions are made for each data point
  - an average of predictions from all the models and use it to make the final prediction.
- Weighted Average
  - an extension of the averaging method.
  - All models are assigned different weights defining the importance of each model for prediction.

Standard ensembling techniques include

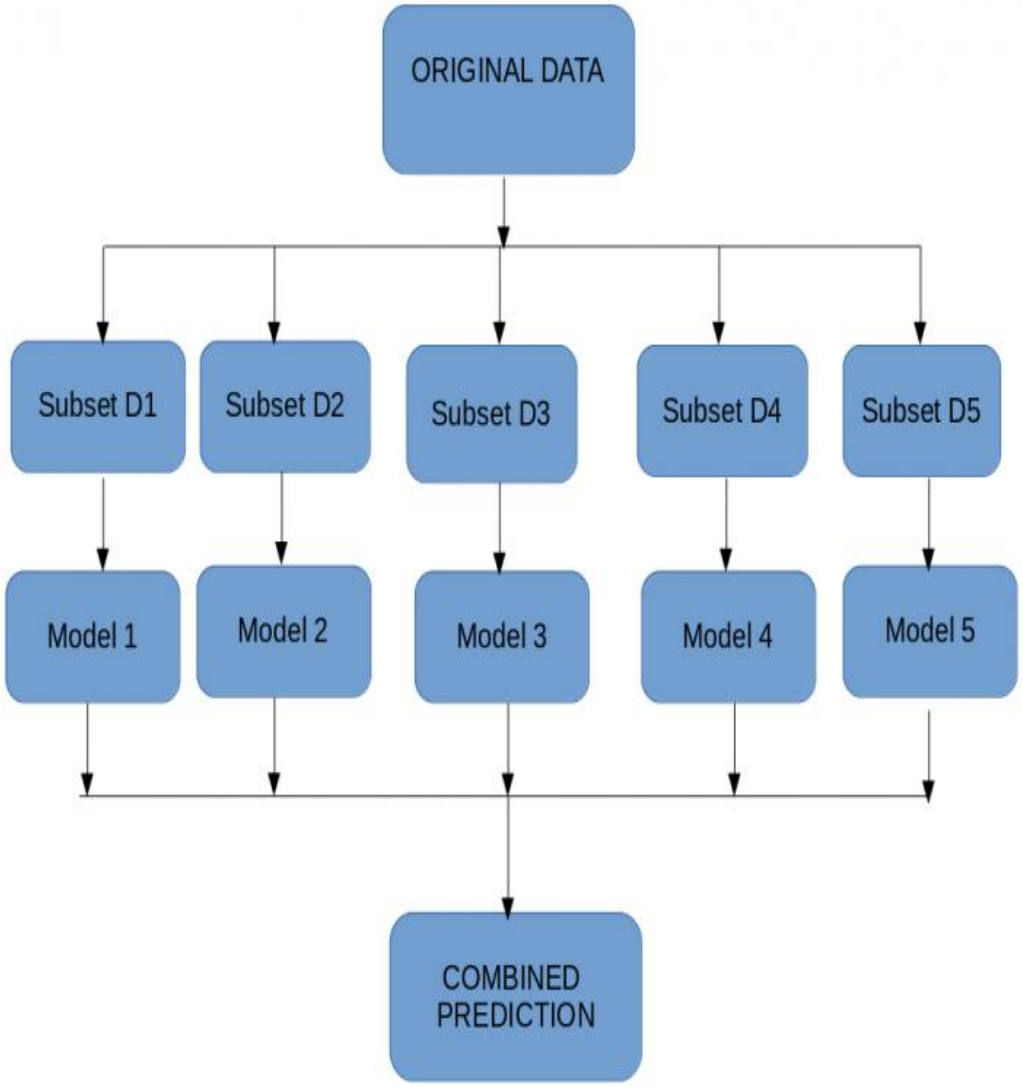
- Bagging
  - involves fitting many decision trees on different samples of the same dataset and averaging the predictions.
- Stacking
  - involves fitting many different models types on the same data
  - and using another model to learn how to best combine the predictions.
- Boosting
  - involves adding ensemble members sequentially that correct the predictions made by prior models and outputs a weighted average of the predictions.

Ensembling in general

- The variance of the general model decreases significantly thanks to **bagging**
- The bias also decreases due to **boosting**
- And overall predictive power improves because of **stacking**
- ***Sequential*** ensemble methods
  - use the dependency between base learners.
  - A popular example of sequential ensemble algorithms is **AdaBoost.**
- ***Parallel*** ensemble methods
  - The base learners are created independently to study
  - Exploit the effects related to their independence and reduce error by averaging the results.
  - An example implementing this approach is **Random Forest**.

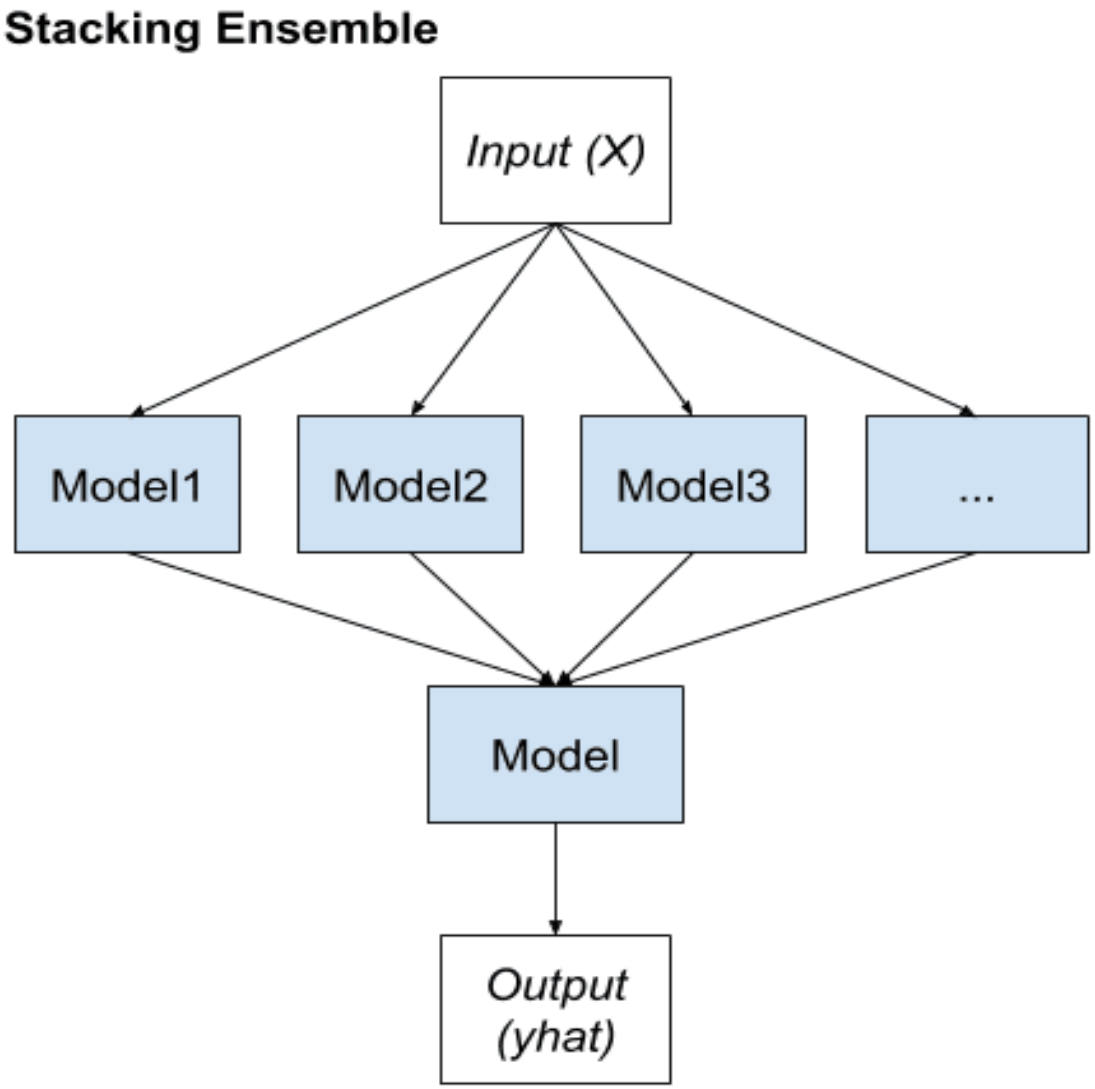
Bagging

- Bootstrapping
  - is a sampling technique in which we create subsets of observations from the original dataset, **with replacement**.
  - The total size of the subsets is the same as the size of the original set.
- Bagging (or Bootstrap Aggregating)
  - uses these subsets (bags) to get a fair idea of the distribution (complete set).
  - The total size of subsets created for bagging may be less than the original set.



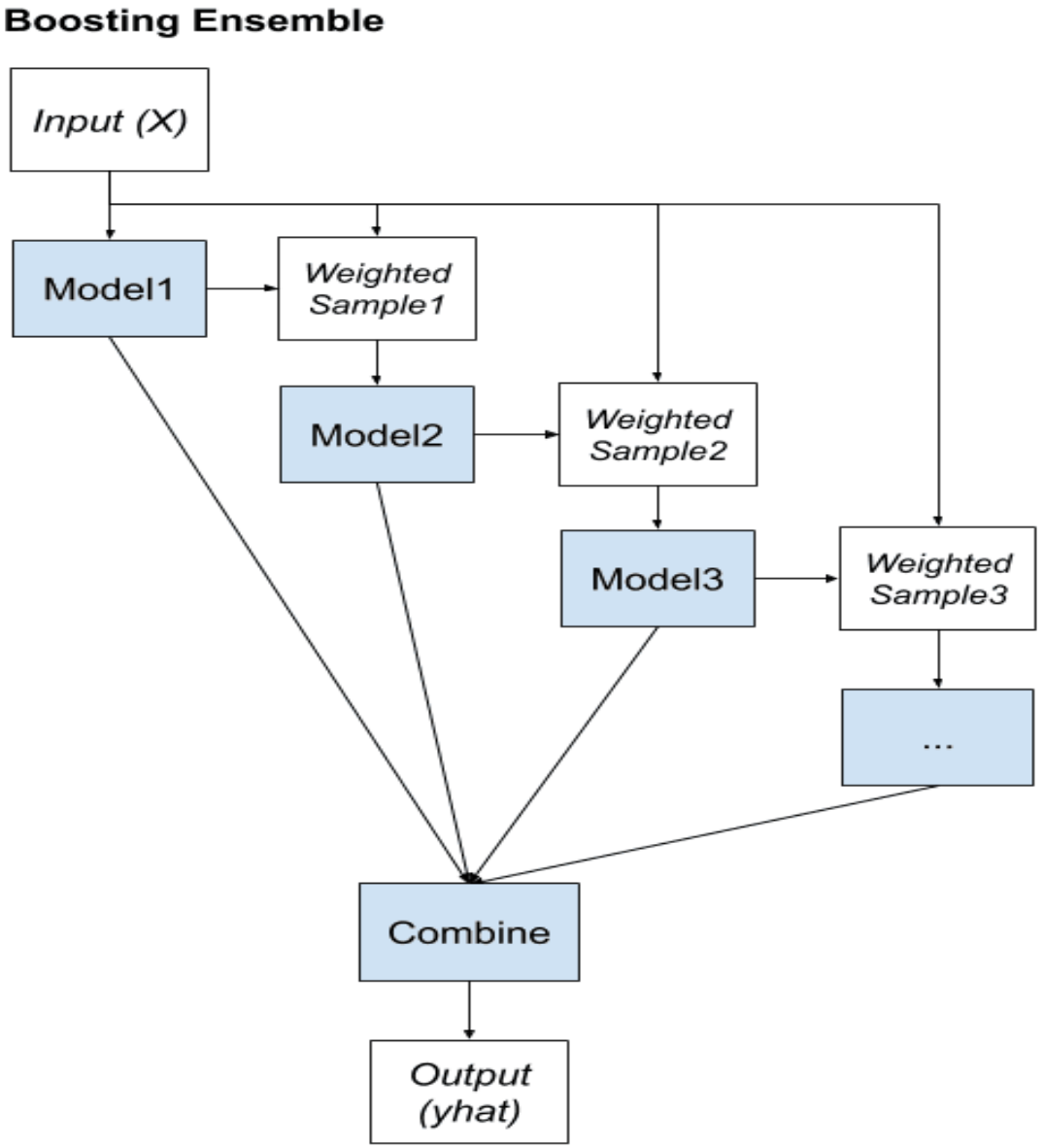
Stacking

- seeks a diverse group of members by varying the model types fit on the training data and using a model to combine predictions.
- level-0 models are ensemble members
- level-1 model is used to combine the predictions



Boosting

- key property is the idea of correcting prediction errors.
- seeks to change the training data to focus attention on examples that previous models were not able to predict
- Steps include :
  - Bias training data toward those examples that are hard to predict.
  - Iteratively add ensemble members to correct predictions of prior models.
  - Combine predictions using a weighted average of models.



Steps in Boosting

1. A subset is created from the original dataset.
2. Initially, all data points are given equal weights.
3. A base model is created on this subset.
4. This model is used to make predictions on the whole dataset.
5. Errors are calculated using the actual values and predicted values.
6. The observations which are incorrectly predicted, are given higher weights.
7. Another model is created and predictions are made on the dataset.
8. Similarly, multiple models are created, each correcting the errors of the previous model.
9. The final model (strong learner) is the weighted mean of all the models (weak learners).

Popular Boosting algorithms

- **Gradient Boost**
  - used in regression and classification tasks
  - prediction model in the form of an ensemble of weak prediction models
  - The target outcome for each instance in the data depends on how much changing that prediction impacts the overall prediction error:
    - If a small change in the prediction for a instance causes a large drop in error, then next target outcome of the instance is a high value.
    - If a small change in the prediction for a instance causes no change in error, then next target outcome of the instance is zero.
  - Usually the base estimator is *Decision Stump*
  - used to minimize bias error of the model.
- **AdaBoost**
  - short for *Adaptive Boosting*
  - is a statistical classification meta-algorithm
  - automatically adjusts its parameters to the data based on the actual performance in the current iteration.
  - Both the weights for re-weighting the data and the weights for the final aggregation are re-computed iteratively.
  - Is adaptive because subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

Popular Boosting algorithms

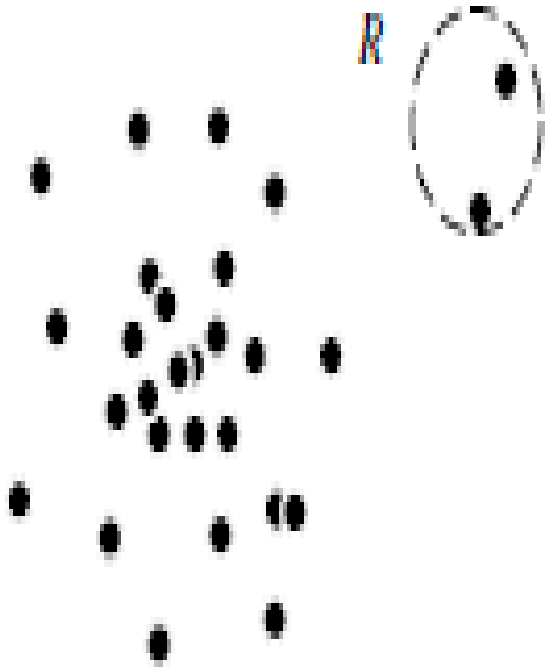
- XG Boost
  - ***Parallelization:*** The model is implemented to train with multiple CPU cores.
  - ***Regularization:*** includes different regularization penalties to avoid overfitting. Penalty regularizations produce successful training so the model can generalize adequately.
  - ***Non-linearity:*** can detect and learn from non-linear data patterns.
  - ***Cross-validation:*** Built-in and comes out-of-the-box.
  - ***Scalability:*** can run distributed thanks to distributed servers and clusters like Hadoop and Spark, so you can process enormous amounts of data.
  - It's also available for many programming languages like C++, JAVA, Python, and Julia.

Anomaly Detection (Outlier Detection)

- is the process of finding data objects with behaviors that are very different from expectation.
- Outlier detection and clustering analysis are two highly related tasks.
  - Clustering finds the majority patterns in a data set and organizes the data accordingly
  - Outlier detection tries to capture those exceptional cases that deviate substantially from the majority patterns
- **Different from Noisy data**
- **Applications include:**
  - **Credit Card Fraud Detection**
  - **Network Intrusion**
- Type of Outliers
  - Global
  - contextual (or conditional)
  - collective

Global Outliers

- a data object is a **global outlier** if it deviates significantly from the rest
- of the data set.
- are sometimes called *point anomalies*
- Example :
  - Intrusion detection in computer networks
  - If the communication behavior of a computer is very different from the normal patterns (e.g., a large number of packages is broadcast in a short time)



Contextual Outliers

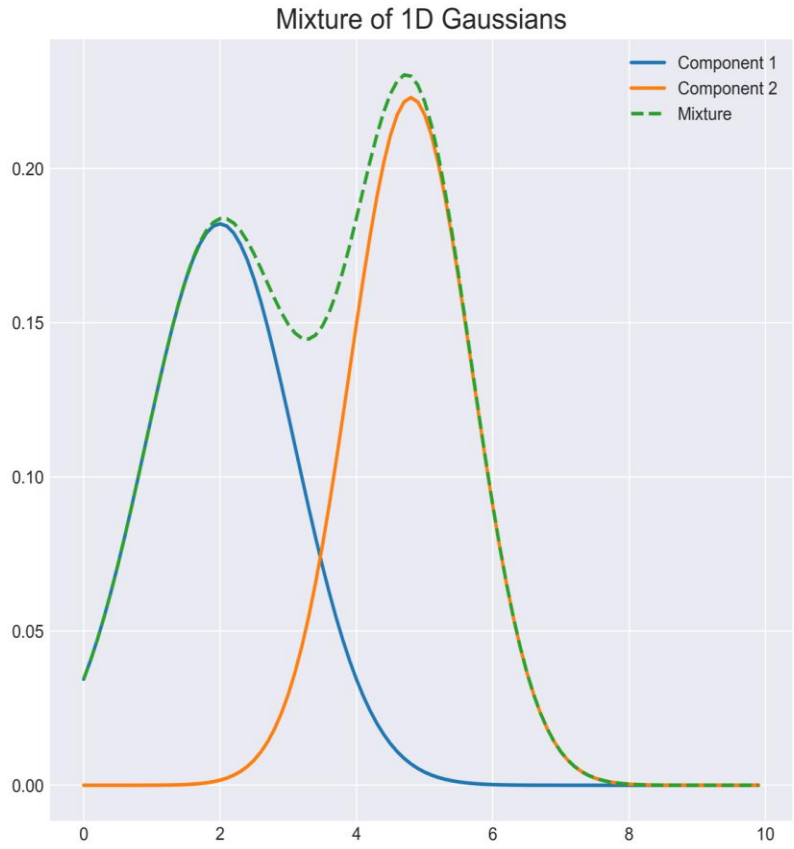
- a data object that deviates significantly with respect to a specific context of the object.
- also known as *conditional outliers* because they are conditional on the selected context.
- divided into two groups:
  - **Contextual attributes**:
    - The contextual attributes of a data object define the object' context.
    - Ex: Temperature based on date and location.
  - **Behavioral attributes**:
    - define the object's characteristics, and are used to evaluate whether the object is an outlier in the context to which it belongs.
    - Ex: Temperature example based on the temperature, humidity, and pressure.

Collective Outliers

- a subset of data objects forms a **collective outlier** if the objects as a whole deviate significantly from the entire data set.
- Importantly, the individual data objects may not be outliers
- to detect background knowledge of the relationship among data objects such as distance or similarity measurements between objects has to be considered
- Example
  - If one shipment is delayed, it may not be considered an outlier
  - However, if 100 orders are delayed on a single day, they can whole form an outlier
- Example
  - intrusion detection
    - a denial-of-service package from one computer to another is considered normal, and not an outlier at all. However, if several computers keep sending denial-of-service packages to each other, they as a whole should be considered as a collective outlier

Outlier Detection Methods

- **Statistical Methods**
  - For a gaussian/normal distribution, the data points lying away from 3rd deviation can be considered as anomalies
- **Histogram-based Outlier Detection**
  - assumes the feature independence and calculates the outlier score by building histograms
  - It is much faster than multivariate approaches, but at the cost of less precision
- **Local Correlation Integral (LOCI)**
  - provides a LOCI plot for each point which summarizes a lot of the information about the data in the area around the point
  - It determines clusters, micro-clusters, their diameters, and their inter-cluster distances

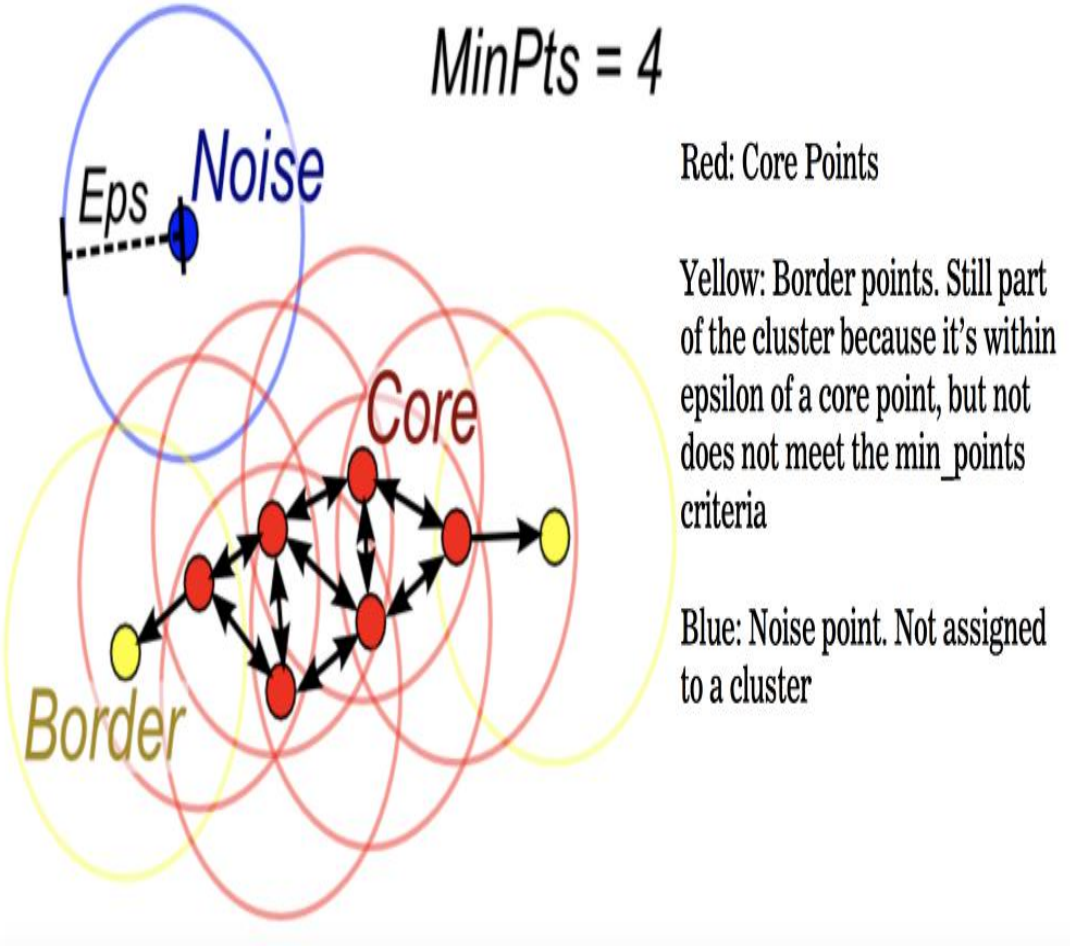


Outlier Detection Methods

- Classification based
  - Consider a training set that contains samples labeled as “normal” and others labeled as “outlier
  - Class imbalance - Number of normal samples likely far exceeds the number of outlier samples.
  - *one-class model*. That is, a classifier is built to describe only the normal class. Any samples that do not belong to the normal class are regarded as outliers.
- **Isolation Forest:**
  - uses a random forest algorithm (decision trees)
  - Tries to split or divide the data points such that each observation gets isolated from the others.
  - the anomalies lie away from the cluster of data points
- **One Class SVM:**
  - regular SVM algorithm tries to find a hyperplane that best separates the two classes of data points.
  - For one-class SVM where we have one class of data points, and the task is to predict a hypersphere that separates the cluster of data points from the anomalies
- K Nearest Neighbor
  - For any data point, the distance to its kth nearest neighbor could be viewed as the outlying score
  - PyOD supports three kNN detectors:
    - Largest: Uses the distance of the kth neighbor as the outlier score
    - Mean: Uses the average of all k neighbors as the outlier score
    - Median: Uses the median of the distance to k neighbors as the outlier score

Outlier Detection Methods

- Clustering based
  - Does the object belong to any cluster? If not, then it is identified as an outlier.
  - Is there a large distance between the object and the cluster to which it is closest? If yes, it is an outlier.
  - Is the object part of a small or sparse cluster? If yes, then all the objects in that cluster are outliers.
- Using Distance Measures
- Using density measures



References

- Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
- Introduction to Machine Learning – Prof. Sudeshna Sarkar,
  - **Week 1:** Introduction: Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation

Next Class..

Supervised Learning: Regression