BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

WORK INTEGRATED LEARNING PROGRAMMES

MACHINE LEARNING MID SEM

Text Book(s)

T1	Tom M. Mitchell, Machine Learning, The McGraw-Hill Companies, Inc. International Edition 1997 **(http://personal.disco.unimib.it/Vanneschi/McGrawHill_-_Machine_Learning_-Tom_Mitchell.pdf)**
T2	Christopher M. Bhisop, Pattern Recognition & Machine Learning, Springer, 2006 **(http://www.rmki.kfki.hu/~banmi/elte/Bishop%20-%20Pattern%20Recognition%20and%20Machine%20Learning.pdf)**

Reference Book(s) & other resources

R1	CHRISTOPHER J.C. BURGES: A Tutorial on Support Vector Machines for Pattern Recognition, Kluwer Academic Publishers, Boston, pp. 1–43.

1. Introduction
   - 1.1. Objective of the course
   - 1.2. Design a Learning System
   - 1.3. Issues in Machine Learning

2. Mathematical Preliminaries
   - 2.1. Linear Algebra, Calculus, Probability theory
   - 2.2. Decision Theory
   - 2.3. Information Theory

3. Bayesian Learning
   - 3.1. MLE Hypothesis
   - 3.2. MAP Hypothesis
   - 3.3. Bayes Rule

4. Linear models for classification
   - 4.1. Probabilistic Generative Classifiers
   - 4.2. Naïve Bayes Classifier
   - 4.3. Discriminant Functions
   - 4.4. Probabilistic Discriminative Classifiers

5. Linear models for Regression

5.1. Linear basis function models
5.2. Bayesian linear regression
5.3. Bias-variance decomposition

6. Decision Tree
   6.1. Avoiding Overfitting
   6.2. Handling Continuous valued attributes, missing attributes
   6.3. Random Forest

Session No.	Topic Title	Study/HW Resource Reference
1	**Introduction** Objective, What is Machine Learning? Application areas of Machine Learning, Why Machine Learning is important? Design a Learning System, Issues in Machine Learning	T1 – Ch1
2	**Mathematical Preliminaries** Linear Algebra, Calculus, Probability theory, Probability Densities, Gaussian Distribution, Decision Theory, Minimum Misclassification Rate, Information Theory, Measure of Information, Entropy	Lecture Notes, T2 – Ch2
3	**Bayesian Learning** MLE Hypothesis, Bayes Rule, MAP Hypothesis, Minimum Description Length (MDL) principle	T1 - Ch. 6
4	**Linear models for classification** Probabilistic Generative Classifiers, Bayes optimal classifier, Naïve Bayes Classifier	T1 - Ch. 6
5	**Linear models for classification** Discriminant Functions, Probabilistic Discriminative Classifiers, text classification model, image classification	T1 – Ch. 6 T2 - Ch. 4
6	**Linear models for Regression** Linear basis function models, Bayesian linear regression, Bias-variance decomposition	T2 - Ch. 3 T1 – Ch. 6
7	**Decision Tree** Handling overfitting, continuous attributes, missing attributes, random forest	T1 – Ch. 3

Week 1 - Machine Learning

- **Machine learning** is a scientific discipline that explores the construction and study of algorithms that can learn from data.
- Such algorithms operate by building a model based on inputs and using that to make predictions or decisions, rather than following only explicitly programmed instructions.

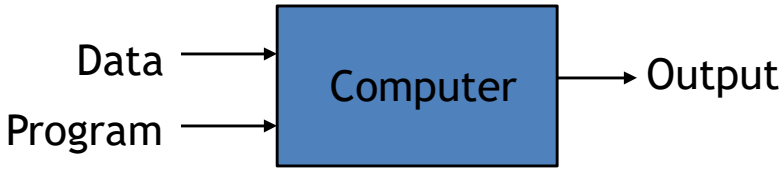
1

A Few Quotes

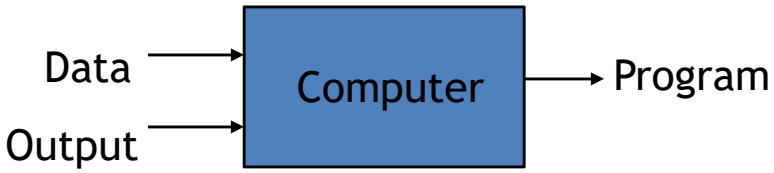
- "A breakthrough in machine learning would be worth ten Microsofts" (Bill Gates, Chairman, Microsoft)
- "Machine learning is the next Internet" (Tony Tether, Director, DARPA)
- "Web rankings today are mostly a matter of machine learning" (Prabhakar Raghavan, Dir. Research, Yahoo)
- "Machine learning is going to result in a real revolution" (Greg Papadopoulos, CTO, Sun)
- "Machine learning is today's discontinuity" (Jerry Yang, CEO, Yahoo)

2

Traditional Programming



Machine Learning



3

Slide credit: Pedro Domingos

3

What is Machine Learning?

Definition by Tom Mitchell (1998):

"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." Example: playing checkers.

E = the experience of playing many games of checkers

T = the task of playing checkers.

P = the probability that the program will win the next game.

4

What is Machine Learning?

- To have a learning problem, we must identify
  - The class of tasks
  - The measure of performance to be improved
  - Source of experience

5

A Checker Learning Problem

- **Task T:** Playing Checkers
- **Performance Measure P:** Percent of games won against opponents
- **Training Experience E:** To be selected ==> Games Played against itself

6

A handwriting recognition learning problem

- **Task T:** recognizing and classifying handwritten words within images
- **Performance measure P**: percent of words correctly classified
- **Training Experience E:** a database of handwritten words with given classifications

7

A robot driving learning problem

- **Task T:** driving on public four-lane highways using vision sensors
- **Performance measure P:** average distance travelled before an error (as judged by human)
- **Training experience E:** a sequence of images and steering commands recorded while observing a human driver

8

Why is Machine Learning Important?

- Some tasks cannot be defined well, except by examples.
- Relationships and correlations can be hidden within large amounts of data. Machine Learning may be able to find these relationships.
- Human designers often produce machines that do not work as well as desired in the environments in which they are used.

9

Why is Machine Learning Important ?

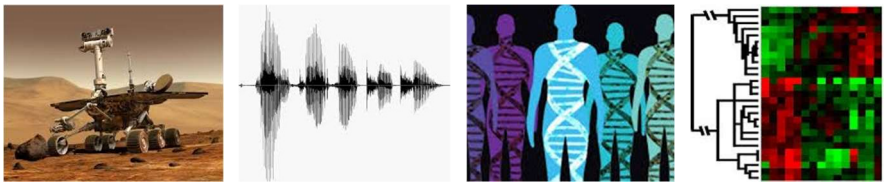
- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans (e.g., medical diagnostic).
- New knowledge about tasks is constantly being discovered by humans. It may be difficult to continuously re-design systems “by hand”.

10

When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



Learning isn't always useful:

- There is no need to "learn" to calculate payroll

11

Slide Credit: Eric Eaton

BITS Pilani, Pilani Campus

11

Applications of Machine Learning

- Learning to recognize spoken words (Lee, 1989; Waibel, 1989).
- Detect fraudulent use of credit cards or Learning to drive an autonomous vehicle (Pomerleau, 1989).
- Learning to classify new astronomical structures (Fayyad et al., 1995).

12

22 December 2019

BITS Pilani, Pilani Campus

12

Applications of Machine Learning

- Learning to play world-class backgammon (Tesauro 1992, 1995).
- Predict recovery rates of pneumonia patients (Copper et al. 1997)

13

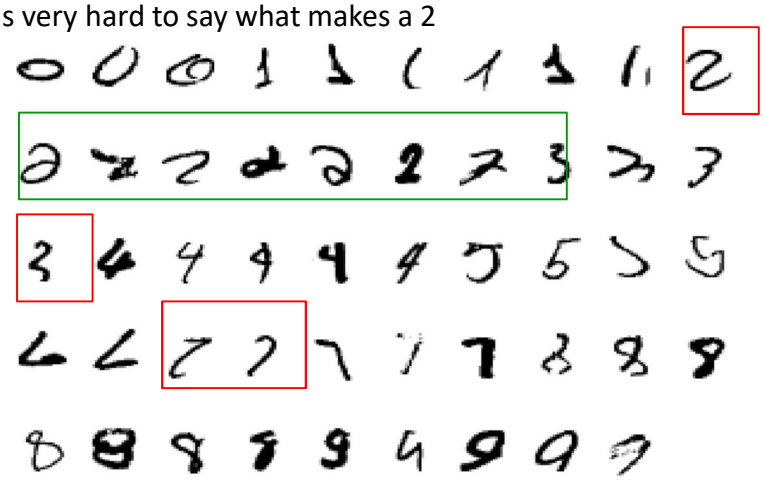
Application Types: Classification

- Medical diagnosis
- Credit card applications or transactions
- Fraud detection in e-commerce
- Worm detection in network packets
- Spam filtering in email
- Recommended articles in a newspaper
- Recommended books, movies, music, or jokes
- Financial investments
- DNA sequences
- Spoken words
- Handwritten letters
- Astronomical images

14

Pattern recognition

It is very hard to say what makes a 2



15

Slide credit: Geoffrey Hinton

BITS Pilani, Pilani Campus

15

Application Domains

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Language Processing

Many more emerging…

16

Slide credit: Pedro Domingos

BITS Pilani, Pilani Campus

16

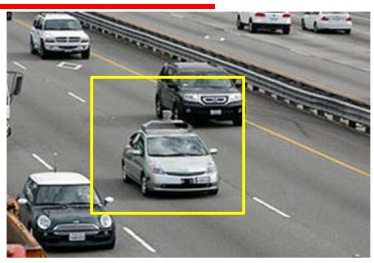
innovate achieve lead

State of the Art Applications of Machine Learning

In this course, you will learn principles that will help you understand and build some of these applications.

innovate achieve lead

Autonomous Cars

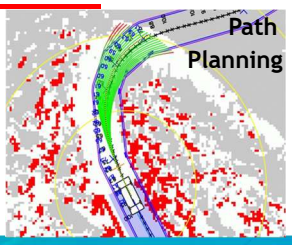


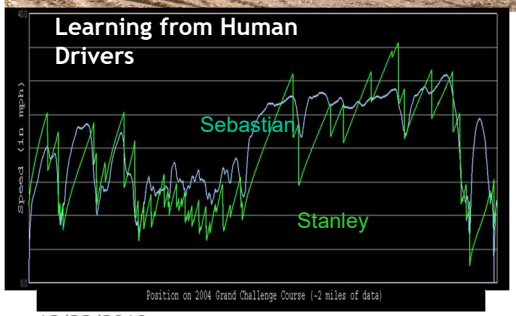
- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

UPenn's Autonomous Car →

Slide credit: Eric Eaton

Autonomous Car Technology





Slide credit: Eric Eaton

19

Deep Learning in the Headlines

BUSINESS NEWS

MIT Technology Review

Is Google Cornering the Market on Deep Learning?

A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 29, 2014

How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.

This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in

BloombergBusinessweek Technology

Acquisitions

The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

intelligence projects. "DeepMind is bona fide in terms of its research capabilities and depth," says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. "We would have more if the talent was there to

WIRED GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY

INNOVATION INSIGHTS community content featured

Deep Learning's Role in the Age of Robots

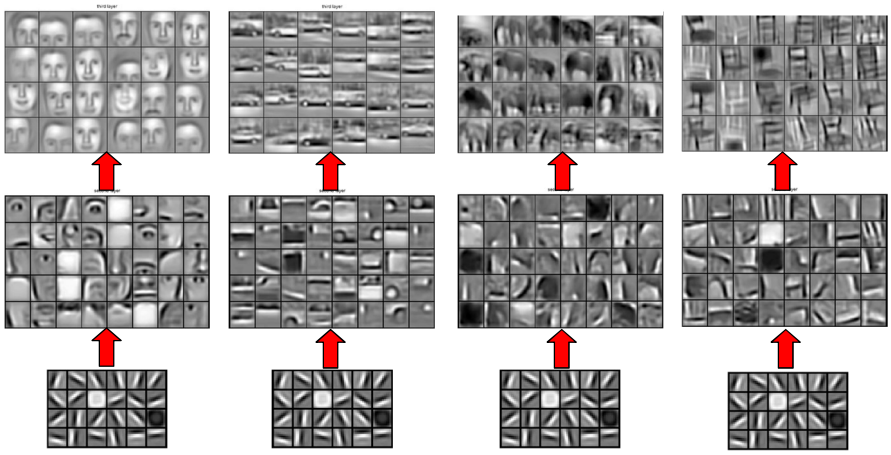
BY JULIAN GREEN, JETPAC 05.02.14 2:56 PM

Slide credit: Eric Eaton

20

innovate achieve lead

Learning of Object Parts



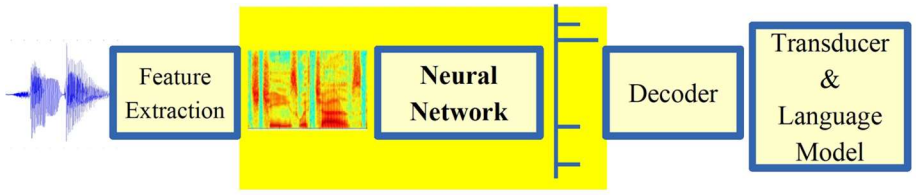
21

Slide credit: Andrew Ng

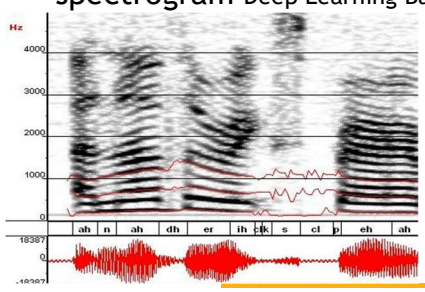
21

innovate achieve lead

Automatic Speech Recognition



ML used to predict phoneme states from sound spectrogram Deep Learning Based Results



# Hidden Layers	1	2	4	8	10	12
Word Error Rate %	16.0	12.8	11.4	10.9	11.0	11.1

Baseline Gaussian Mixture Model based word error rate = 15.4%

[Zeiler et al. "On rectified linear units for speech recognition" ICASSP 2013]

22

Slide credit: Eric Eaton

22

Types of Learning

- **Supervised (inductive) learning**
  - Given: training data, desired outputs (labels)
- **Unsupervised learning**
  - Given: training data (without desired outputs)
- **Semi-supervised learning**
  - Given: training data + a few desired outputs
- **Reinforcement learning**
  - Given: rewards from sequence of actions

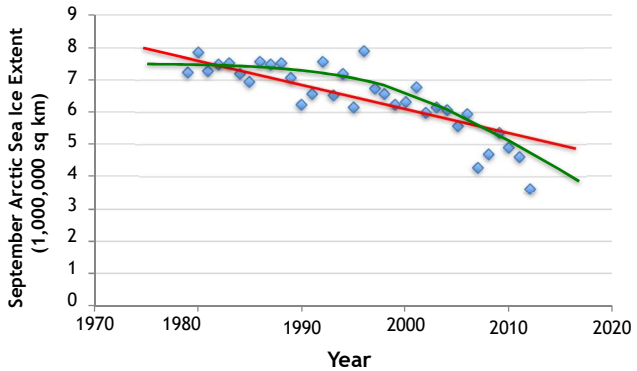
23

Slide Credit: Eric Eaton

23

Supervised Learning: Regression

- Given (x_1, y_1), (x_2, y_2), …, (x_n, y_n)
- Learn a function $f(x)$ to predict y given x



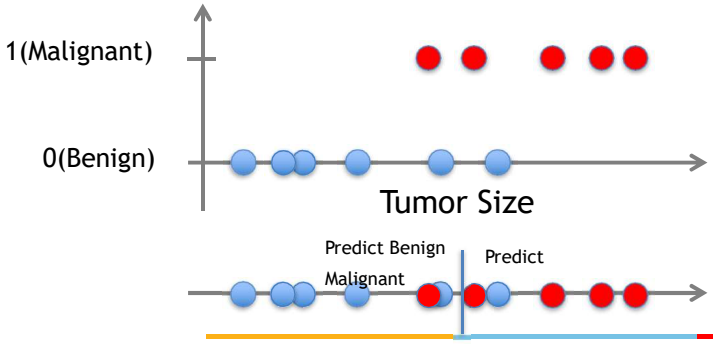
Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013) Slide Credit: Eric Eaton 24

24

Supervised Learning: Classification

- Given (x_1, y_1), (x_2, y_2), …, (x_n, y_n)
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification

Breast Cancer (Malignant / Benign)



25

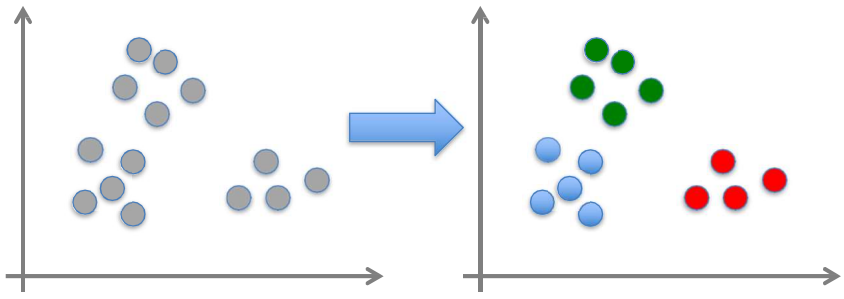
Slide Credit: Eric Eaton

BITS Pilani, Pilani Campus

25

Unsupervised Learning

- Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's
  - E.g., clustering



26

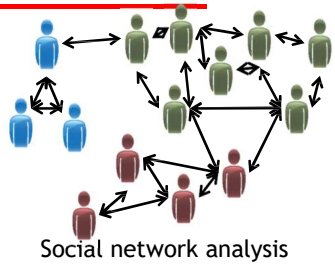
Slide Credit: Eric Eaton

BITS Pilani, Pilani Campus

26

Unsupervised Learning

Organize computing clusters



Social network analysis

Market segmentation

Astronomical data analysis

27

Slide credit: Andrew Ng

27

Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states → actions that tells you what to do in a given state
- Examples:
  - Credit assignment problem
  - Game playing
  - Robot in a maze

28

Slide Credit: Eric Eaton

28

innovate achieve lead

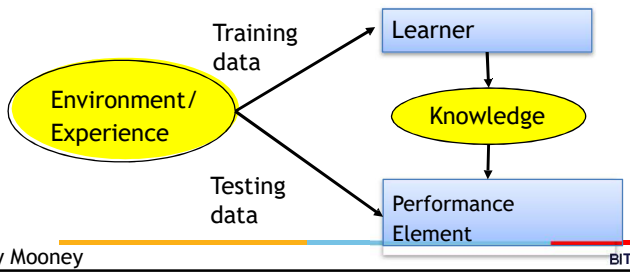
Design a Learning System

29

innovate achieve lead

Designing a Learning System

- Choose the training experience(data)
- Choose exactly what is to be learned
  - i.e. the ***target function***
- Choose how to represent the target function
- Choose a learning algorithm to infer the target function from the experience



Slide Credit: Ray Mooney

30

Designing a Learning System: An Example

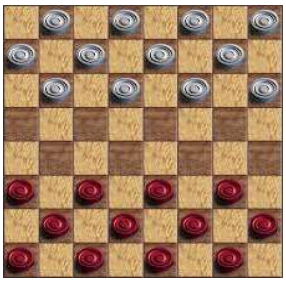
1. Problem Description (Ex: Playing checkers)
2. Choosing the Training Experience (data expressed as features)
3. Choosing the Target Function to be learnt (Ex: deciding next board position
4. Choosing a Representation for the Target Function (design a function as linear etc)
5. Choosing a Function Approximation Algorithm (parameters learning using loss function)
6. Final Design

Choosing the training experience

- The first problem to choose the type of training experience from which our system will learn.
  - ✓ The type of training experience available can have a significant impact on success or failure of the learner.
  - ✓ One key attribute is whether the training experience provides ***direct*** or ***indirect*** feedback regarding the choices made by the performance system.

Choosing the training experience

- In learning to play checkers, the system might learn from **direct training** examples consisting of individual checkers board states and the correct move for each.
- Alternatively, it might have available only **indirect information** consisting of the move sequences and final outcomes of various games played.



Choosing the training experience

- In **indirect training**, information about the correctness of specific moves early in the game must be inferred indirectly from the fact that the game was eventually won or lost.
- The learner faces an additional problem of **credit assignment**, or determining the degree to which each move in the sequence deserves credit or blame for the final outcome.
- Credit assignment can be a particularly difficult problem because the game can be lost even when early moves are optimal, if these are followed later by poor moves. Hence, learning from direct training feedback is typically easier than learning from indirect feedback.

Choosing the training experience

- A second important attribute of the training experience is the degree to which the learner controls the sequence of training examples.
  - the learner might rely on the teacher to select informative board states and to provide the correct move for each.
  - the learner might itself propose board states that it finds particularly confusing and ask the teacher for the correct move.

35

Choosing the training experience

  - the learner may have complete control over both the board states and (indirect) training classifications, as it does when it learns by playing against itself with no teacher present.
    - the learner may choose between experimenting with novel board states that it has not yet considered, or sharpen its skill by playing minor variations of lines of play it currently finds most promising.

36

Choosing the training experience

- A third important attribute of the training experience is how well it represents the distribution of examples over which the final system performance ***P*** must be measured.
  - learning is most reliable when the training examples follow a distribution similar to that of future test examples.
  - the performance metric ***P*** is the percent of games the system wins in the world tournament. If its training experience ***E*** consists only of games played against itself, there is an obvious danger that this training experience might not be fully representative of the distribution of situations over which it will later be tested.

Choosing the training experience

- For example, the learner might never encounter certain crucial board states that are very likely to be played by the human checkers champion.
- it is often necessary to learn from a distribution of examples that is somewhat different from those on which the final system will be evaluated
- one distribution of examples will not necessary lead to strong performance over some other distribution.

innovate achieve lead

Choosing the Target Function

- *ChooseMove*, however, is difficult to learn.
- An easier and related target function to learn is function

V: B --> R, which assigns a numerical score to each board. The better the board positions B, the higher the score R.

- If the system can successfully learn such a target function ***V,*** then it can easily use it to select the best move from any current board position.
- This can be accomplished by generating the successor board state produced by every legal move, then using ***V*** to choose the best successor state and therefore the best legal move.

39

innovate achieve lead

Choosing the Target Function

- Let us therefore define the target value ***V(b)*** for an arbitrary board state ***b*** in ***B***, as follows:
  1. If ***b*** is a final board state that is won, then ***V(b) = 100***
  2. If ***b*** is a final board state that is lost, then ***V(b) = -100***
  3. If ***b*** is a final board state that is draw, then ***V(b) = 0***
  4. If ***b*** is a not a final state in the game, then ***V(b) = V(b')***, where ***b'*** is the best final board state that can be achieved starting from ***b*** and playing optimally until the end of the game.

40

innovate achieve lead

Choosing the Target Function

- This recursive definition specifies a value of ***V(b)*** for every board state ***b.***
- It is not usable by player - it is not efficiently computable - it is a ***nonoperational definition.***
- The goal of learning in this case is to discover an ***operational description of V*** i.e., a description that can be used by the checkers - playing program to evaluate states and select moves within realistic time bounds.

41

innovate achieve lead

Choosing the Target Function

- Reduced the learning task to the problem of discovering an operational description of the ideal target function ***V*** – it is difficult to learn ***V*** perfectly.
- We often expect learning algorithms to acquire only some ***approximation*** to the target function -- is called ***function approximation.*** [The actual function can often not be learned and must be approximated]

42

Choosing a Representation for the Target Function

- **Expressiveness versus Training set size**
  - More expressive the representation of the target function, the closer to the "truth" we can get.
  - More expressive the representation, the more training examples are necessary to choose among the large number of "representable" possibilities.
- **Example of a representation:**
  - x1/x2 = # of black/red pieces on the board
  - x3/x4 = # of black/red king on the board
  - x5/x6 = # of black/red pieces threatened by red/black

$$\hat{V}(b) = w0+w1.x1+w2.x2+w3.x3+w4.x4+w5.x5+w6.x6$$

wi's are adjustable or "learnable" coefficients

Choosing a Representation for the Target Function

- w0 through w6 are numerical coefficients, or weights, to be chosen by the learning algorithm.
- Learned values for the weights w1 through w6 will determine the relative importance of the various board features in determining the value of the board, whereas the weight w0 will provide an additive constant to the board value.

Choosing a Representation for the Target Function

Partial design of a checkers learning program:

- Task T: playing checkers
- Performance measure P: percent of games won in the world tournament
- Training experience E: games played against itself
- *Target function:* $V: Board \rightarrow \Re$
- *Target function representation*

$$\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

Choosing a Function Approximation Algorithm

- **Generating Training Examples of the form <b,V_{train}(b)>** [e.g. <x1=3, x2=0, x3=1, x4=0, x5=0, x6=0, +100 (=blacks won)]
  - x1= # of black pieces on the board
  - x2 = # of red pieces on the board
  - x3 = # of black king on the board
  - x4 = # of red king on the board
  - x5 = # of black pieces threatened by red
  - x6 = # of red pieces threatened by black

Choosing a Function Approximation Algorithm

- **Estimating the training Values**
  - only training information available to learner is whether the game was eventually won or lost.
  - we require training examples that assign specific scores to specific board states.
  - easy to assign a value to board states that correspond to the end of the game,
  - less obvious how to assign training values to the more numerous intermediate board states that occur before the game's end.
  - fact that the game was eventually won or lost does not necessarily indicate that every board state along the game path was necessarily good or bad.

47

Choosing a Function Approximation Algorithm

- **Estimating the training Values**
  - Despite the ambiguity inherent in estimating training values for intermediate board states.
  - one simple approach is to assign the training value of $V_{train}(b)$ for **any intermediate board** state **b** to be $\hat{V}$**(Successor(b))**, where $\hat{V}$ is the learner's current approximation to **V** and where **Successor(b)** denotes the next board state following **b**
  - This rule for estimating training values can be summarized as

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

48

innovate achieve lead

Choosing a Function Approximation Algorithm

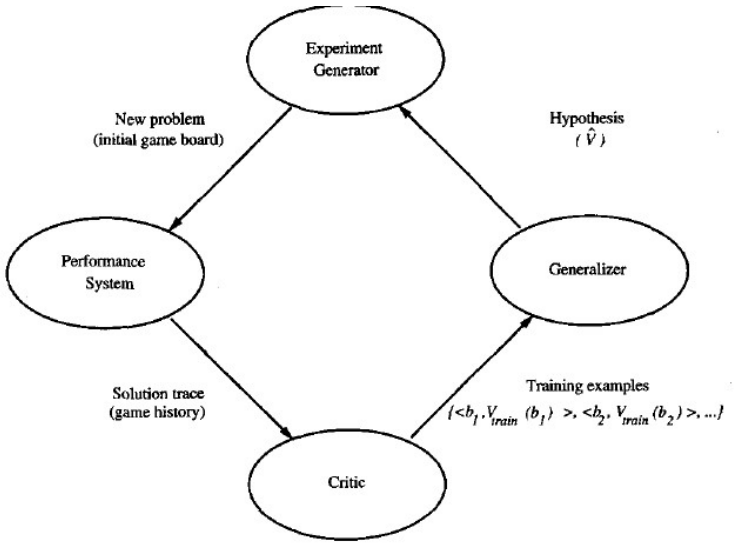
- **Adjust the weights**
  - Specify the learning algorithm for choosing the weights **wi** to best fit the set of training examples {**<b,V_{train}(b)>** }.
  - One common approach is to define the best hypothesis, or set of weights, as that which minimizes the squared error ***E*** between the training values and the values predicted by the hypothesis $\hat{V}$.

$$E \equiv \sum_{\langle b, V_{train}(b)\rangle \in \text{ training examples}} (V_{train}(b) - \hat{V}(b))^2$$

  - Thus, we seek the weights, or equivalently the $\hat{V}$, that minimize ***E*** for the observed training examples.

innovate achieve lead

Final Design for Checkers Learning



Final Design for Checkers Learning

- **The Performance Module (performance)** : Takes as input a new board and outputs a trace of the game it played against itself.
- **The Critic (data generation)** : Takes as input the trace of a game and outputs a set of training examples of the target function
- **The Generalizer (learner)**: Takes as input training examples and outputs a hypothesis which estimates the target function.
- **The Experiment Generator (task):** Takes as input the current hypothesis (currently learned function) and outputs a new problem (an initial board state) for the performance system to explore

Issues in Machine Learning

- What algorithms are available for learning a concept? How well do they perform?
- How much training data is sufficient to learn a concept with high confidence?
- When is it useful to use prior knowledge?
- Are some training examples more useful than others?
- What are the best tasks for a system to learn?
- What is the best way for a system to represent its knowledge?

ML in a Nutshell

innovate achieve lead

- Tens of thousands of machine learning algorithms
  - Hundreds new every year
- Every ML algorithm has three components
  - **Representation**
  - **Optimization**
  - **Evaluation**

53

Slide credit: Pedro Domingos

53

Regression

innovate achieve lead

$$y = f(\mathbf{x})$$

output — prediction function — features

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

54

Slide credit: L. Lazebnik

54

Regression Example

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

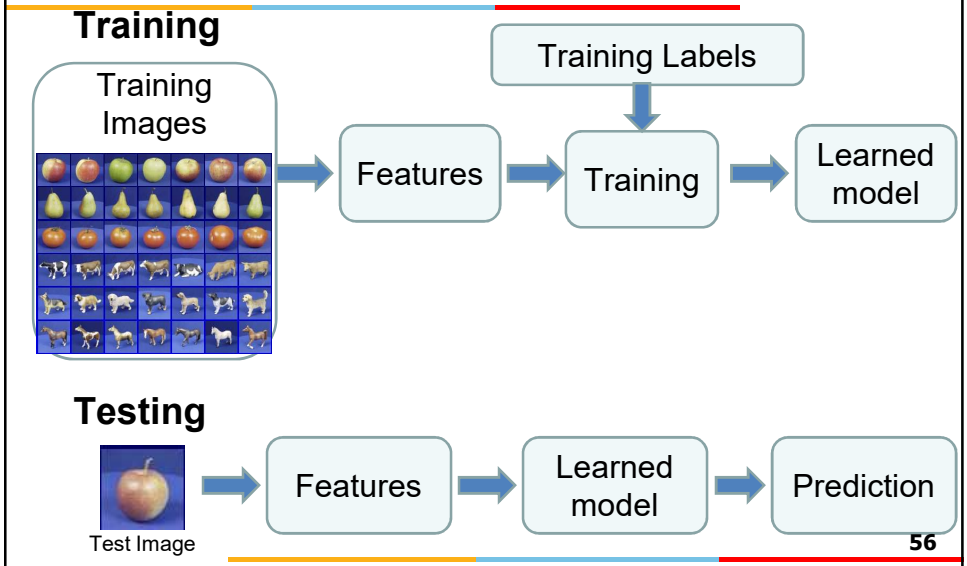
f() = “cow”

55

Slide credit: L. Lazebnik

55

Classification



56

Slide credit: D. Hoiem and L. Lazebnik

56

Function Representations

- Numerical functions
  - Linear regression
  - Neural networks
  - Support vector machines
- Symbolic functions
  - Decision trees
  - Rules in propositional logic
  - Rules in first-order predicate logic
- Instance-based functions
  - Nearest-neighbor
  - Case-based
- Probabilistic Graphical Models
  - Naïve Bayes
  - Bayesian networks
  - Hidden-Markov Models (HMMs)
  - Probabilistic Context Free Grammars (PCFGs)
  - Markov networks

Slide credit: Ray Mooney

57

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- etc.

Slide credit: Pedro Domingos

58

Evaluating Performance

- If *y* is discrete:
  - Accuracy: # correctly classified / # all test examples
  - Precision/recall
    - True Positive, False Positive, True Negative, False Negative
    - **P**recision = TP / (TP + FP) = # predicted true pos / # predicted pos
    - **R**ecall = TP / (TP + FN) = # predicted true pos / # true pos
  - F-measure
    = 2PR / (P + R)
- Want evaluation metric to be in some range, e.g. [0 1]
  - 0 = worst possible classifier, 1 = best possible classifier

59

Evaluating Performance

- If *y* is continuous:
  - Sum-of-Squared-Differences (SSD) error between predicted and true *y*:

$$\mathbf{E} = \sum_{i=1}^{n} (\mathbf{f}(\mathbf{x_i}) - \mathbf{y_i})^2$$

60

Training vs Testing

- What do we want?
  - High accuracy on training data?
  - No, high accuracy on *unseen/new/test data!*
  - Why is this tricky?
- Training data
  - Features (x) and labels (y) used to learn mapping f
- Test data
  - Features used to make a prediction
  - Labels only used to see how well we've learned f!!!
- Validation data
  - Held-out set of the *training data*
  - Can use both features and labels to tune *parameters* of the model we're learning

Slide Credit: Adriana

Training vs. Test Distribution

- We generally assume that training and test examples are independently drawn from the same overall distribution of data
  - We call this "i.i.d" which stands for "independent and identically distributed"

Slide credit: Ray Mooney

ML in Practice

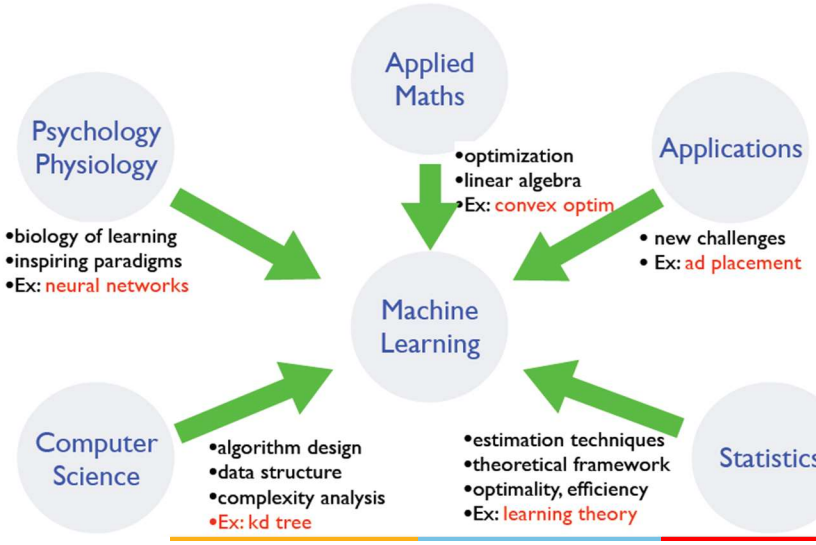
- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing, etc.
- Learn models
- Interpret results
- Consolidate and deploy discovered knowledge

63

Based on a slide by Pedro Domingos

63

Where does ML fit in?



64

Slide credit: Dhruv Batra, Fei Sha

64

WEEK 2 – MATH PRELIMS

- Linear Algebra Review
- Calculus Review
- Probability Theory (Ref: 1.2)
- Decision Theory (Ref: 1.5)
- Information Theory (Ref: 1.6)

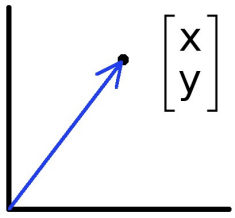
65

Vectors and Matrices

- Collections of ordered numbers that represent movements in space, scaling factors, word counts, movie ratings, pixel brightness, etc.
- Vector is a mathematical quantity that has magnitude and direction

66

Vectors



- Vectors can represent an offset in 2D or 3D space
- Points are just vectors from the origin

- Data can also be treated as a vector
- Such vectors don't have a geometric interpretation, but calculations like "distance" still have value

67

Vector

- A column vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$ where

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- A row vector $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$ where

$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

T denotes the transpose operation

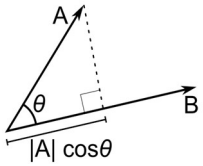
68

Inner Product

- Multiply corresponding entries of two vectors and add up the result

$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \qquad \text{(scalar)}$$

- If B is a unit vector, then A·B gives the length of A which lies in the direction of B (projection)



(if B is unit-length hence norm is 1)

69

Norms

- **Norm** is a function that assigns a strictly positive *length* or *size* to each vector in a vector space—except for the zero vector
- **L^1 norm -** One-dimensional vector spaces

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$$

- **L^2 norm -** *n*-dimensional Euclidean space $\mathbf{R}^n$,

$$\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}$$

- **L^p norm** - Let p ≥ 1 be a real number. The p norm of vector $x=(x_1, x_2,\dots x_n)$

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

70

Matrix

- A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an array of numbers with size $m \downarrow$ by $n \rightarrow$, i.e. m rows and n columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

- If $m = n$, we say that $\mathbf{A}$ is square.

71

Matrix Operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

  - Can only add a matrix with matching dimensions, or a scalar.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

72

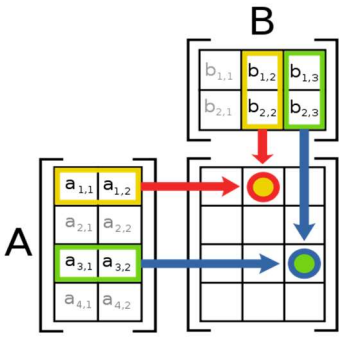
Matrix Multiplication

- Let X be an *a*x*b* matrix, Y be an *b*x*c* matrix
- Then Z = X*Y is an *a*x*c* matrix
- Second dimension of first matrix, and first dimension of first matrix have to be the same, for matrix multiplication to be possible

73

Matrix Multiplication

- The product AB is:



- Each entry in the result is (that row of A) dot product with (that column of B)

74

Different types of product

- ***x, y*** = column vectors (nx1)
- ***X, Y*** = matrices (mxn)
- *x, y* = scalars (1x1)

- $\boldsymbol{x}^T \boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{y}$ = inner product (1xn x nx1 = scalar)
- $\boldsymbol{x} \otimes \boldsymbol{y} = \boldsymbol{x}\,\boldsymbol{y}^T$ = outer product (nx1 x 1xn = matrix)

- ***X* Y*** = matrix product
- ***X .* Y*** = element-wise product

75

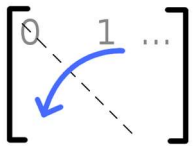
Inverse

- Given a matrix **A**, its inverse $\mathbf{A}^{-1}$ is a matrix such that $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
- E.g. $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$
- Inverse does not always exist. If $\mathbf{A}^{-1}$ exists, **A** is *invertible* or *non-singular*. Otherwise, it's *singular*.

76

Matrix Operations

- Transpose – flip matrix, so row 1 becomes column 1



$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- A useful identity:

$$(ABC)^T = C^T B^T A^T$$

77

Matrix Rank

- Column/row rank
  - col-rank(A)=no. of independent columns
  - row-rank(A)=no. of independent rows
- Column rank always equals row rank
- Matrix rank $\text{rank}(\mathbf{A}) \triangleq \text{col-rank}(\mathbf{A}) = \text{row-rank}(\mathbf{A})$
- If a matrix is not full rank, inverse doesn't exist
  - Inverse also doesn't exist for non-square matrices

78

Matrix Operation Properties

- Matrix addition is commutative and associative
  - A + B = B + A
  - A + (B + C) = (A + B) + C
- Matrix multiplication is associative and distributive but *not* commutative
  - A(B*C) = (A*B)C
  - A(B + C) = A*B + A*C
  - A*B != B*A

79

Linear independence

- Suppose we have a set of vectors $\mathbf{v_1}, \ldots, \mathbf{v_n}$
- If we can express $\mathbf{v_1}$ as a linear combination of the other vectors $\mathbf{v_2}\ldots\mathbf{v_n}$, then $\mathbf{v_1}$ is linearly *dependent* on the other vectors.
  - The direction $\mathbf{v_1}$ can be expressed as a combination of the directions $\mathbf{v_2}\ldots\mathbf{v_n}$. (E.g. $\mathbf{v_1}$ = .7 $\mathbf{v_2}$ -.7 $\mathbf{v_4}$)
- If no vector is linearly dependent on the rest of the set, the set is linearly *independent*.
  - Common case: a set of vectors $\mathbf{v_1}, \ldots, \mathbf{v_n}$ is always linearly independent if each vector is perpendicular to every other vector (and non-zero)

80

Singular Value Decomposition (SVD)

- There are several computer algorithms that can "factor" a matrix, representing it as the product of some other matrices
- The most useful of these is the Singular Value Decomposition
- **Singular value decomposition** (**SVD**) is a factorization of a real or complex matrix
- Represents any matrix **A** as a product of three matrices: $\mathbf{U\Sigma V^T}$

https://en.wikipedia.org/wiki/Singular_value_decomposition

81

Singular Value Decomposition (SVD)

- In general, if **A** is *m* x *n*, then **U** will be *m* x *m*, **Σ** will be *m* x *n*, and $\mathbf{V^T}$ will be *n* x *n*.

$$\underset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \underset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \underset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \underset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

82

Singular Value Decomposition (SVD)

For square, positive semi-definite matrix A

$$\mathbf{U\Sigma V^T = A}$$

- Where **U** and **V** are rotation matrices, and **Σ** is a scaling matrix.

$$\underset{U}{\begin{bmatrix}-.40 & .916\\ .916 & .40\end{bmatrix}} \times \underset{\Sigma}{\begin{bmatrix}5.39 & 0\\ 0 & 3.154\end{bmatrix}} \times \underset{V^T}{\begin{bmatrix}-.05 & .999\\ .999 & .05\end{bmatrix}} = \underset{A}{\begin{bmatrix}3 & -2\\ 1 & 5\end{bmatrix}}$$

- U and V are unitary matrices, i.e., $UU^T = U^TU = I$ (identity matrix)
- Orthogonal matrix is a square matrix whose columns and rows are orthogonal unit vectors (i.e., orthonormal vectors), i.e. $Q^TQ=QQ^T=I$, where I is the identity matrix.
- Matrix Q is orthogonal if its transpose is equal to its inverse: $Q^T=Q^{-1}$.

https://www.youtube.com/watch?v=NsNNI_-JPUY

83

Positive-semidefinite matrix

- Symmetric ($M = M^T$) $n * n$ real matrix M is said to be **positive-semidefinite** if the scalar $x^T M x$ is positive or zero for every non-zero column vector x of n real numbers

(domain of x = 1*n, domain of M= n*n and x^T = n*1)

- When interpreting $M x$ as the output of matrix, M, that is acting on an input, x, the property of **positive-semidefinite** implies that the output always is positive or zero ; inner product with the input

$$M \text{ positive semi-definite} \iff x^T Mx \geq 0 \text{ for all } x \in \mathbb{R}^n$$

84

innovate achieve lead

Differentiation

The derivative provides us information about the rate of change of a function.

The derivative of a function is also a function.

Example:

The derivative of the acceleration function is the velocity function.

85

innovate achieve lead

Derivative = rate of change

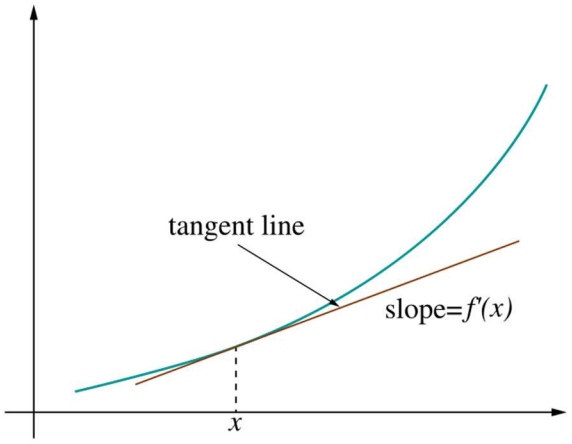


Image: Wikipedia

86

innovate achieve lead

Derivative = rate of change

- Linear function y = mx + b
- Slope $$m = \frac{\text{change in } y}{\text{change in } x} = \frac{\Delta y}{\Delta x},$$

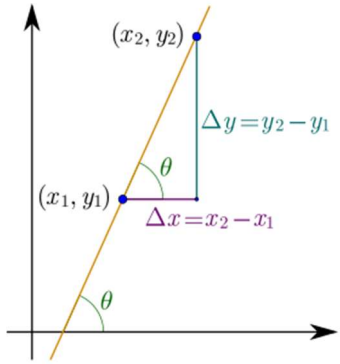


Image: Wikipedia

BITS Pilani

87

innovate achieve lead

Ways to Write the Derivative

Given the function f(x), we can write its derivative in the following ways:

- f'(x)
- $\frac{d}{dx}$f(x)

The derivative of x is commonly written dx.

Texas A&M Dept of Statistics

BITS Pilani

88

Differentiation Formulas

The following are common differentiation formulas:

- The derivative of a constant is 0.

$$\frac{d}{du}c = 0$$

- The derivative of a sum is the sum of the derivatives.

$$\frac{d}{du}(f(u) + g(u)) = f'(u) + g'(u)$$

89

More Formulas

- The derivative of *u* to a constant power:

$$\frac{d}{du}u^n = n * u^{n-1}$$

- The derivative of *e*:

$$\frac{d}{du}e^u = e^u$$

- The derivative of *log*:

$$\frac{d}{du}\log(u) = \frac{1}{u}$$

90

innovate achieve lead

Product and Quotient

The product rule and quotient rules are commonly used in differentiation.

- Product rule:

$$\frac{d}{du}(f(u)*g(u)) = f(u)g'(u) + g(u)f'(u)$$

- Quotient rule:

$$\frac{d}{du}\left(\frac{f(u)}{g(u)}\right) = \frac{g(u)f'(u) - f(u)g'(u)}{g^{2}(u)}$$

innovate achieve lead

Chain Rule

The chain rule allows you to combine any of the differentiation rules we have already covered.

- First, do the derivative of the outside and then do the derivative of the inside.

$$\frac{d}{du}f(g(u)) = f'(g(u))*g'(u)*du$$

innovate achieve lead

Try These

$f(z) = z + 11$

$s(y) = 4ye^{2y}$

$g(y) = 4y^3 + 2y$

$p(x) = \dfrac{\log(x^2)}{x}$

$h(x) = e^{3x}$

$q(z) = (e^z - z)^3$

93

innovate achieve lead

Solutions

$f'(z) = 1$

$s'(y) = 8ye^{2y} + 4e^{2y}$

$g'(y) = 12y^2 + 2$

$p'(x) = \dfrac{2 - \log(x^2)}{x^2}$

$h'(x) = 3e^{3x}$

$q'(z) = 3(e^z - z)^2(e^z - 1)$

94

innovate achieve lead

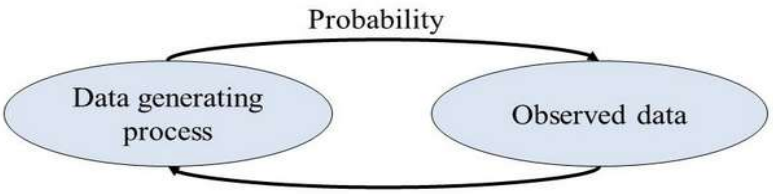
Probability Review

BITS Pilani

95

innovate achieve lead

Probability Theory



- **Probability Theory**
  - Given a data generating process, what are the properties of the outcome?
- **Statistical Inference**
  - Given the outcome, what can we say about the process that generated the data?
  - How can we generalize these observations and make predictions about future outcomes?

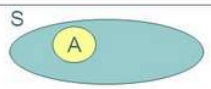
BITS Pilani

96

Probability Theory

Let A be an event, then we denote

$P(A)$ the probability for A



It always hold that $0 \le P(A) \le 1$ $P(\emptyset) = 0$ $P(S) = 1$

Consider an experiment which has N equally likely outcomes, and let exactly n of these events correspond to the event A. Then

$$P(A) = \frac{n}{N} = \frac{\text{\# successful outcomes}}{\text{\# possible outcomes}}$$

Example:
Rolling a dice

$P(\text{even number}) = \frac{3}{6} = \frac{1}{2}$

97

Random Variable

- A **random variable**, usually written X, is a **variable** whose possible values are numerical outcomes of a **random** phenomenon or experiment.

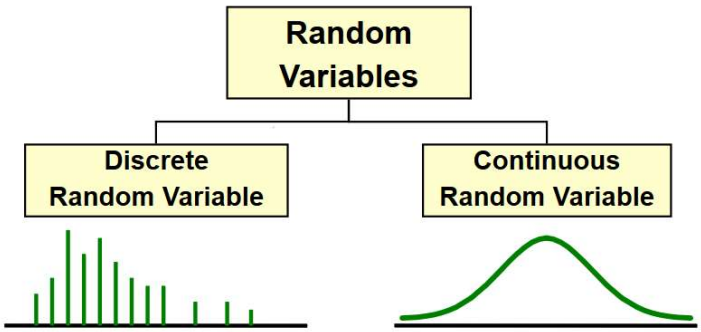
Examples

- ✔ X = number of heads when the experiment is flipping a coin 20 times.
- ✔ C = the daily change in a stock price.
- ✔ R = the number of kilometers per litter you get on your car during a family vacation.

98

Random Variable

Represents a possible numerical value from a random event



99

Random Variable

Discrete Random Variable

- one that takes on a ***countable*** number of values
- usually count data [Number of]
- list **all** possible outcomes without missing any of them

Example:

- ✔ X = sum of values on the roll of two dice:
  X has to be either 2, 3, 4, …, or 12.
- ✔ Y = number of students in MTech DSE:
  Y has to be 60,65,70 ……

100

Random Variable

innovate achieve lead

Continuous Random Variable

- Variable that takes on an uncountable number of values
- Usually measurement data [time, weight, distance, etc]
- You can never list all possible outcomes even if you had an infinite amount of time

Example:

X = time it takes you to drive home from work place: X > 0, might be 30.1 minutes measured to the nearest tenth but in reality the actual time is 30.10000001…… minutes?)

Exercise: try to list all possible numbers between 0 and 1

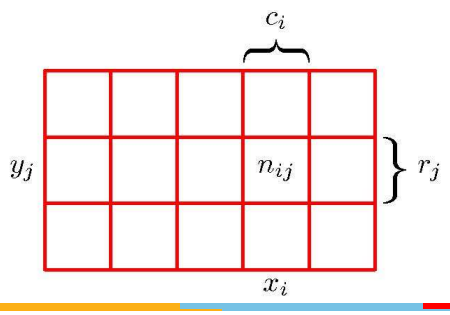
BITS Pilani

101

Probability Theory

innovate achieve lead

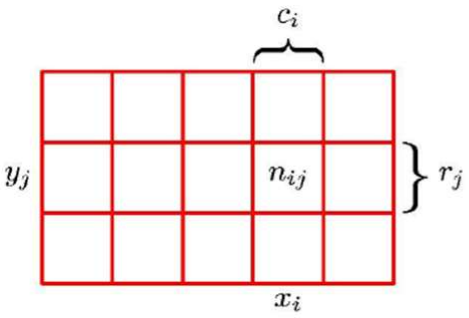
- X and Y are two discrete random variables.
- X can take any of the values x_i, i = 1, . . . , M, and Y can take the values y_j, j = 1, . . . , L.
- Let a total of N trials in which we sample both of the variables X and Y, and let the number of such trials in which X = x_i and Y = y_j be n_{ij} .
- Let the number of trials in which X takes the value x_i be denoted by c_i, and similarly let the number of trials in which Y takes the value y_j be denoted by r_j .



BITS Pilani

102

Probability Theory



Marginal Probability

$$p(X = x_i) = \frac{c_i}{N}.$$

Joint Probability

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

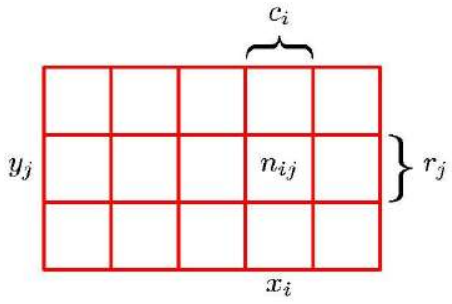
Conditional Probability

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

Here we are implicitly considering the limit $N \to \infty$.

103

Probability Theory



Sum Rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N} \sum_{j=1}^{L} n_{ij}$$

$$= \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Product Rule

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N}$$

$$= p(Y = y_j | X = x_i) p(X = x_i)$$

104

Probability Theory

Notation

- A **random variable X** represents outcomes or states of the world.
- We will write p(x) to mean Probability(X = x)
- **Sample space**: the space of all possible outcomes (may be discrete, continuous, or mixed)
- p(x) is the **probability mass (density) function**
  - Assigns a number to each point in sample space
  - Non-negative, sums (integrates) to 1
  - Intuitively: how often does x occur, how much do we believe in x.

Probability Theory

Joint Probability Distribution

- Prob(X=x, Y=y)
  - "Probability of X=x and Y=y"
  - p(x, y)

Conditional Probability Distribution

- Prob(X=x|Y=y)
  - "Probability of X=x given Y=y"
  - p(x|y) = p(x,y)/p(y)

Probability Theory

The Rules of Probability

- Sum Rule (marginalization/summing out):

$$p(x) = \sum_{y} p(x, y)$$

$$p(x_1) = \sum_{x_2} \sum_{x_3} \dots \sum_{x_N} p(x_1, x_2, \dots, x_N)$$

- Product/Chain Rule:

$$p(x, y) = p(y \mid x) p(x)$$

$$p(x_1, \dots, x_N) = p(x_1) p(x_2 \mid x_1) \dots p(x_N \mid x_1, \dots, x_{N-1})$$

107

Probability Theory

from the definition of conditional distributions:

$$P(A|B)P(B) = P(A, B) = P(B|A)P(A)$$

Hence:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

is known as Bayes rule.

example:

$$P(\text{“taking a shower”}|\text{“wet”}) = P(\text{“wet”}|\text{“taking a shower”}) \frac{P(\text{“taking a shower”})}{P(\text{“wet”})}$$

$$P(\text{reason}|\text{observation}) = P(\text{observation}|\text{reason}) \frac{P(\text{reason})}{P(\text{observation})}$$

108

Probability Theory

innovate achieve lead

Bayes rule - Example

if patient has meningitis, then very often a stiff neck is observed

if patient has meningitis, then very often a stiff neck is observed
$P(S|M) = 0.8$ (can be easily determined by counting)

observation: 'I have a stiff neck! Do I have meningitis?' (is it reasonable to be afraid?)

109

Probability Theory

innovate achieve lead

Bayes rule - Example

if patient has meningitis, then very often a stiff neck is observed
$P(S|M) = 0.8$ (can be easily determined by counting)

observation: 'I have a stiff neck! Do I have meningitis?' (is it reasonable to be afraid?)
$P(M|S) = ?$

we need to now: $P(M) = 0.0001$ (one of 10000 people has meningitis)
and $P(S) = 0.1$ (one out of 10 people has a stiff neck).

110

Probability Theory

if patient has meningitis, then very often a stiff neck is observed
$P(S|M) = 0.8$ (can be easily determined by counting)

observation: 'I have a stiff neck! Do I have meningitis?' (is it reasonable to be afraid?)
$P(M|S) = ?$

we need to now: $P(M) = 0.0001$ (one of 10000 people has meningitis)
and $P(S) = 0.1$ (one out of 10 people has a stiff neck).

then:

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$$

Keep cool. Not very likely

111

Probability Densities

Continuous Probability Distribution

Let X be a continuous rv. Then a *probability distribution or probability density function (pdf)* of X is a function $f(x)$ such that for any two numbers a and b,

$$P(a \le X \le b) = \int_a^b f(x)dx$$

The graph of f is the *density curve*.

112

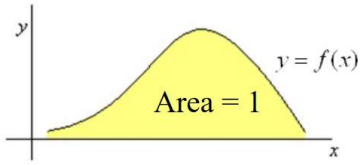
Probability Densities

Probability Density Function

For $f(x)$ to be a pdf

1. $f(x) > 0$ for all values of x.

2. The area of the region between the graph of f and the $x-$ axis is equal to 1.

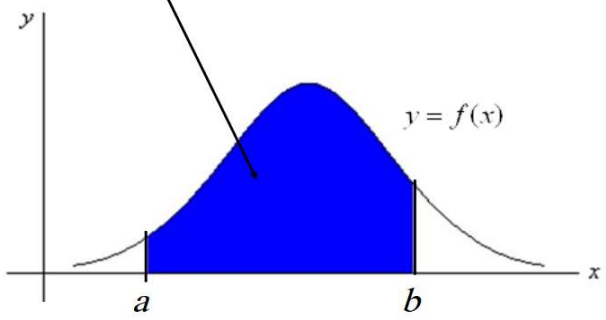


113

Probability Densities

Probability Density Function

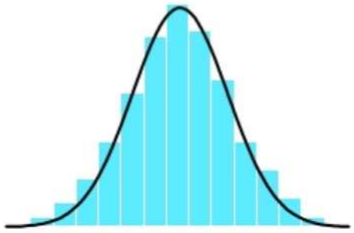
$P(a \leq X \leq b)$ is given by the area of the shaded region.



114

Gaussian Distribution

- The commonest and the most useful continuous distribution.
- A symmetrical probability distribution where most results are located in the middle and few are spread on both sides.
- It has the shape of a bell.
- Can entirely be described by its mean and standard deviation.



115

Gaussian Distribution

Examples:

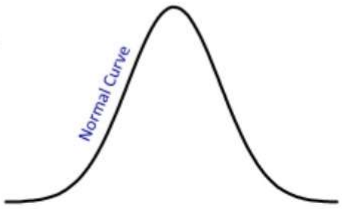
- The body temperature for healthy humans.
- The heights and weights of adults.
- The thickness and dimensions of a product.
- IQ and standardized test scores.
- Quality control test results.
- Errors in measurements.

116

Gaussian Distribution

Normal Curve:

- A graphical representation of the normal distribution.
- It is determined by the mean and the standard deviation.
- It a symmetric unimodal bell-shaped curve.
- Its tails extending infinitely in both directions.
- The wider the curve, the larger the standard deviation and the more variation exists in the process.
- The spread of the curve is equivalent to six times the standard deviation of the process.

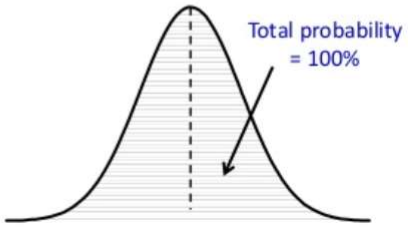


117

Gaussian Distribution

Normal Curve:

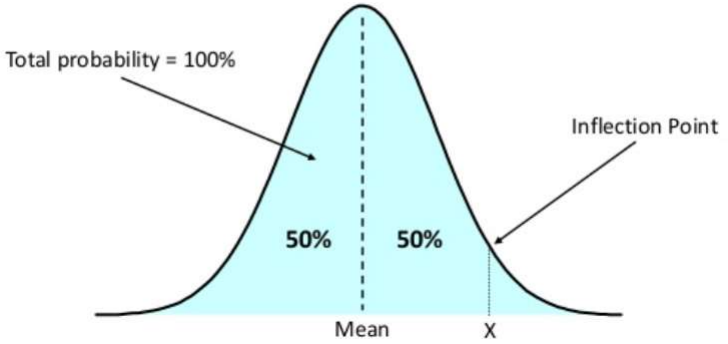
- Helps calculating the probabilities for normally distributed populations.
- The probabilities are represented by the area under the normal curve.
- The total area under the curve is equal to **100%** (or **1.00**).
- This represents the population of the observations.
- We can get a rough estimate of the probability above a value, below a value, or between any two values.



118

Gaussian Distribution

- Since the normal curve is symmetrical, **50 percent** of the data lie on each side of the curve.

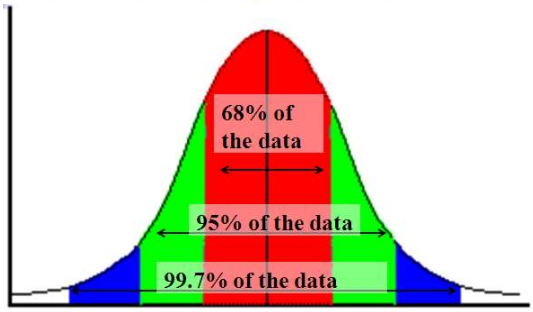


119

Gaussian Distribution

Empirical Rule:

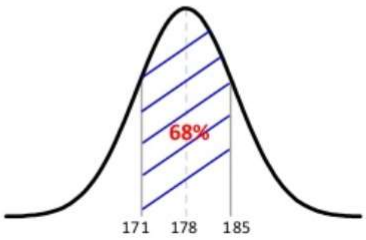
- For any normally distributed data:
  - **68%** of the data fall within **1** standard deviation of the mean.
  - **95%** of the data fall within **2** standard deviations of the mean.
  - **99.7%** of the data fall within **3** standard deviations of the mean.



120

Gaussian Distribution

- Suppose that the heights of a sample men are normally distributed.
- The mean height is **178** cm and a standard deviation is **7** cm.
- **We can generalize that:**
  - **68%** of population are between **171** cm and **185** cm.
  - This might be a generalization, but it's true if the data is normally distributed.



121

Mean, Variance & Standard Deviation

- ✔ The mean of a discrete random variable is the ***weighted average*** of all of its values. The weights are the probabilities.
- ✔ This parameter is also called the expected value of X and is represented by E(X).

$$E(X) = \mu = \sum_{all\ x} xP(x)$$

- ✔ The variance is

$$V(X) = \sigma^2 = \sum_{all\ x} (x-\mu)^2 P(x)$$

- ✔ The standard deviation is

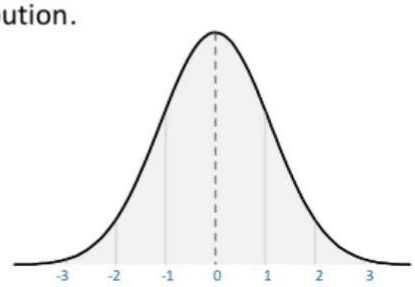
$$\sigma = \sqrt{\sigma^2}$$

122

Gaussian Distribution

Standard Normal Distribution:

- A common practice to convert any normal distribution to the standardized form and then use the standard normal table to find probabilities.
- The **Standard Normal Distribution** (Z distribution) is a way of standardizing the normal distribution.
- It always has a mean of **0** and a standard deviation of **1**.



123

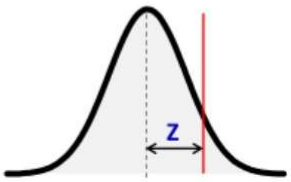
Gaussian Distribution

Standard Normal Distribution:

- Any normally distributed data can be converted to the standardized form using the formula:

$$Z = (X - \mu) / \sigma$$

- where:
  - 'X' is the data point in question.
  - 'Z' (or **Z-score**) is a measure of the number of standard deviations of that data point from the mean.

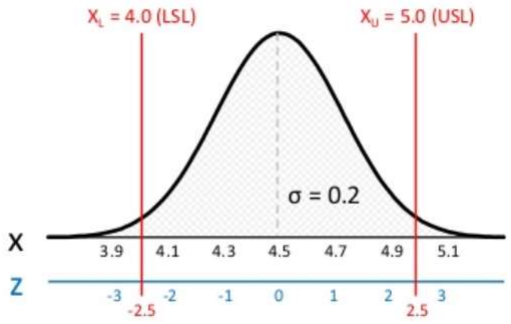


124

Gaussian Distribution

Standard Normal Distribution:

- Converting from 'X' to 'Z':



The specification limits at 4.0 and 5.0mm respectively, and the corresponding z values

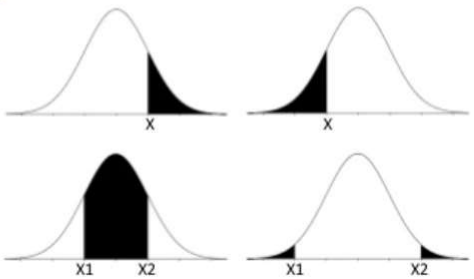
The X-scale is for the actual values and the Z-scale is for the standardized values

125

Gaussian Distribution

Standard Normal Distribution:

- You can then use this information to determine the area under the normal distribution curve that is:
  - To the right of your data point.
  - To the left of the data point.
  - Between two data points.
  - Outside of two data points.



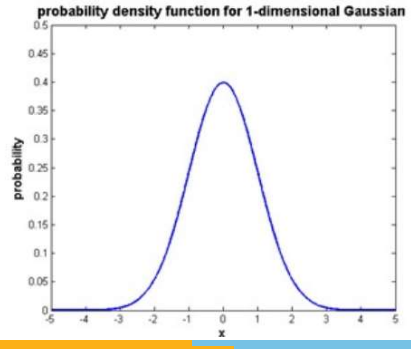
126

innovate achieve lead

Gaussian Distribution

In one dimension

$$N(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$



BITS Pilani

127

innovate achieve lead

Gaussian Distribution

In one dimension

$$N(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Causes pdf to decrease as distance from center increases

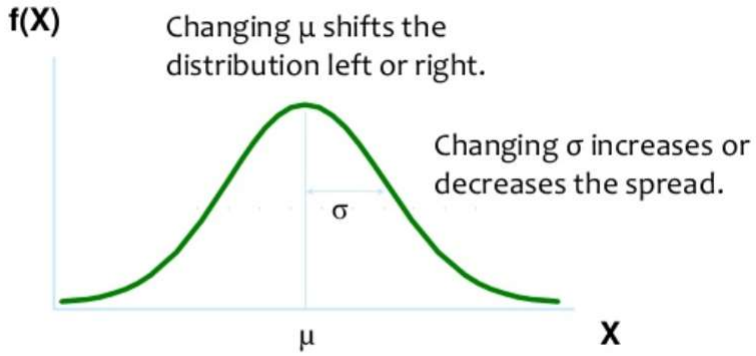
Controls width of curve

Normalizing constant: insures that distribution integrates to 1

BITS Pilani

128

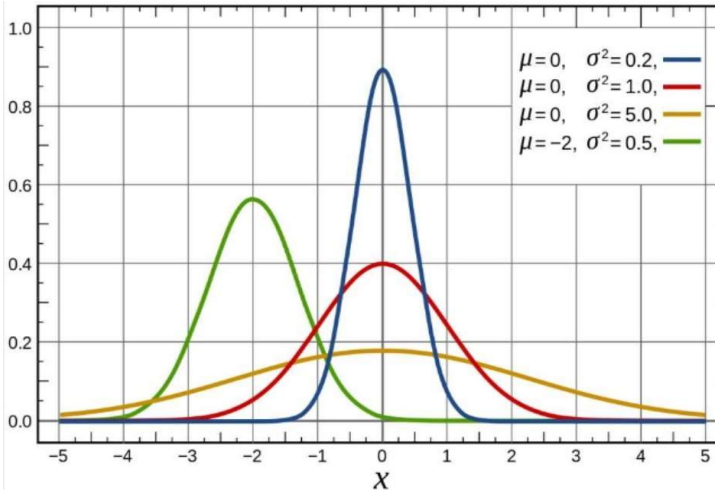
Gaussian Distribution



The normal curve is not a single curve but a family of curves, each of which is determined by its mean and standard deviation.

129

Gaussian Distribution



130

Multivariate Gaussian Distribution

In *d* dimensions

$$N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

x and μ now *d*-dimensional vectors

- μ gives center of distribution in *d*-dimensional space

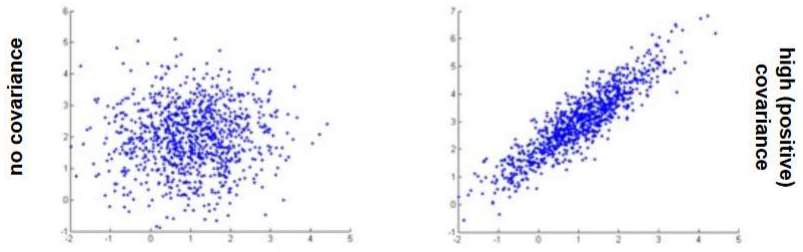
σ^2 replaced by Σ, the *d* x *d* covariance matrix

- Σ contains pairwise covariances of every pair of features
- Diagonal elements of Σ are variances σ^2 of individual features
- Σ describes distribution's shape and spread

Multivariate Gaussian Distribution

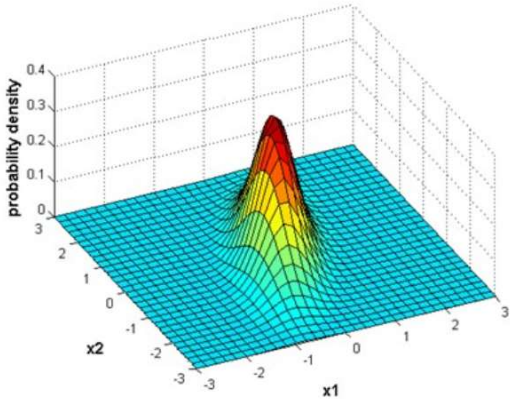
Covariance

- Measures tendency for two variables to deviate from their means in same (or opposite) directions at same time



Multivariate Gaussian Distribution

In two dimensions



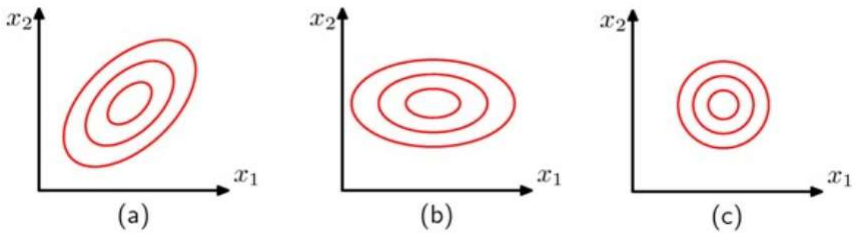
$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.25 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

133

Multivariate Gaussian Distribution

In two dimensions



$$\boldsymbol{\Sigma} = \begin{bmatrix} 2 & 0.6 \\ 0.6 & 2 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

134

innovate achieve lead

Decision Theory

135

innovate achieve lead

Decision Theory

- Suppose **x** is an input vector together with a corresponding vector **t** of target variables
- Goal: predict **t** given a new value for **x.**
- The joint probability distribution ***p(x, t)*** provides a complete summary of the uncertainty associated with these variables.
- Determination of ***p(x, t)*** from a set of training data is called ***inference*** and is a difficult problem.

136

Decision Theory

Inference step

Determine either $p(t|\mathbf{x})$ or $p(\mathbf{x}, t)$.

Decision step

For given $\mathbf{x}$, determine optimal t.

137

Example : Medical diagnosis problem

Input: X-ray image of a patient
Input vector x is the set of pixel intensities in the image

Output: Presence of cancer = Class C1,
Absence of cancer, = Class C2.
Choose t to be a binary variable such that
t =0 corresponds to C1 and t =1 corresponds to C2.

We are interested in the probabilities of the two classes given the image, which are given by $p(C_k|x)$.
Using Bayes' theorem,

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

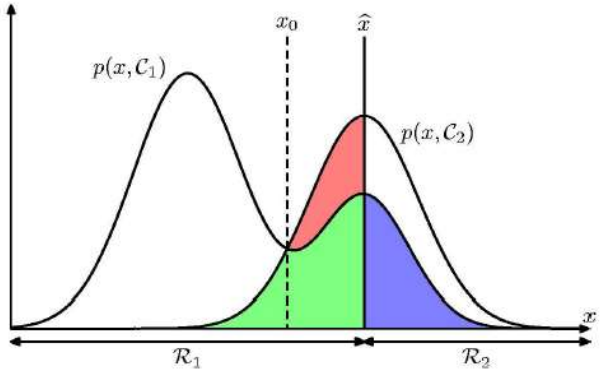
138

Minimum Misclassification Rate

- Divide the input space into regions Rk called decision regions, one for each class, such that all points in Rk are assigned to class Ck
- Boundaries between decision regions are called decision boundaries or decision surfaces
- A mistake occurs when an input vector belonging to class C1 is assigned to class C2 or vice versa.

139

Minimum Misclassification Rate



$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, \mathrm{d}\mathbf{x}.
\end{aligned}
$$

140

Minimum Misclassification Rate

$$p(\text{correct}) = \sum_{k=1}^{K} p(\mathbf{x} \in \mathcal{R}_k, \mathcal{C}_k)$$
$$= \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(\mathbf{x}, \mathcal{C}_k)\, d\mathbf{x}$$

141

Inference and Decision

- We have broken the **classification problem** down into two separate stages, the ***inference stage*** in which we use training data to learn a model for $p(C_k|x)$, and the subsequent ***decision stage*** in which we use these posterior probabilities to make optimal class assignments.
- An alternative possibility would be to solve both problems together and simply learn a function that maps inputs ***x*** directly into decisions. Such a function is called a ***discriminant function***.
- **Three distinct approaches to solving decision problems**

142

Inference and Decision

1st approach

- Determine the class-conditional densities ***p(x | C_k)*** for each class ***C_k*** individually.
- Separately infer the prior class probabilities ***p(C_k)*** . Then use Bayes' theorem

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

143

Inference and Decision

- Equivalently, we can model the joint distribution ***p(x, C_k)*** directly and then normalize to obtain the posterior probabilities ***p(C_k |x)***.
- Use decision theory to determine class membership for each new input ***x***.
- Approaches that explicitly or implicitly model the distribution of inputs as well as outputs are known as ***generative models***, because by sampling from them it is possible to generate synthetic data points in the input space.

144

Inference and Decision

2nd approach

- Solve the inference problem of determining the posterior class probabilities $p(C_k|x)$, and then subsequently use decision theory to assign each new x to one of the classes.
- Approaches that model the posterior probabilities directly are called ***discriminative models***.

145

Inference and Decision

3rd approach

- Find a function $f(x)$, called a ***discriminant function***, which maps each input x directly onto a class label.
- Example - For two-class problems, $f(\cdot)$ might be binary valued and such that $f = 0$ represents class C_1 and $f = 1$ represents class C_2. In this case, probabilities play no role.

146

Inference and Decision

- 1st approach is the most demanding because it involves finding $p(x,C_k)$.
- For many applications, x will have high dimensionality, and consequently we may need a large training set in order to be able to determine the class-conditional densities to reasonable accuracy.
- The class priors $p(C_k)$ can often be estimated simply from the fractions of the training set data points in each of the classes.

147

Inference and Decision

- One advantage of 1st approach, however, is that it also allows **p(x)** to be determined. This can be useful for detecting new data points that have low probability under the model and for which the predictions may be of low accuracy, which is known as ***outlier detection*** or ***novelty detection.***

$$p(\mathrm{x}) = \sum_k p(\mathrm{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

148

Inference and Decision

- If we only wish to make classification decisions, then it can be wasteful of computational resources, and excessively demanding of data, to find $p(x, C_k)$ when in fact we only really need the posterior probabilities $p(C_k|x)$, which can be obtained directly through 2nd approach.

149

Inference and Decision

- The 3rd approach is simple, where the data is used to find a ***discriminant function f(x)*** that maps each ***x*** directly onto a class label, thereby combining the inference and decision stages into a single learning problem.

150

Information Theory
Measure of Information

- It rained heavily in Shillong yesterday
- There was a heavy rainfall in Rajasthan last night.
- The amount of information conveyed by a message is inversely proportional to its probability of occurrence. That is

$$I_k \alpha \frac{1}{p_k}$$

151

Measure of Information

- The information conveyed by a message cannot be negative. It has to be at least 0, i.e.,

$$I_k \geq 0$$

- If $p_k = 1$, then $I_k = 0$.
- The information conveyed composite statement which is independent is simply given by the sum of the individual self-information contents, i.e.,

$$I(m_1, m_2) = I(m_1) + I(m_2)$$

152

Measure of Information

- The only mathematical operator satisfies above properties is the logarithmic operator. Therefore,

$$I_k = \log_r \frac{1}{p_k} \; units$$

- If r = 2, unit is bits

153

Average Information Content (Entropy)

- Consider a source x = {x_1, x_2, ..., x_N}, with probabilities P = {p_1, p_2, ..., p_N}
- A message of length L emitted by the source, then it contains
- p_1 x L number of symbol x_1
- p_2 x L number of symbol x_2, ...,
- p_N x L number of symbol x_N

154

Average Information Content (Entropy)

- The self-information of x_1 is

$$I_{x_1} = \log_2 \frac{1}{p_1} \ bits$$

- Each symbol x_1 conveys information of $\log_2(1/p_1)$ bits and such (p_1 x L) number of x_1 symbols are present on an average in a length of L symbols.
- The total information conveyed by symbols of type x_i is

$$p_i \times L \times \log_2 \frac{1}{p_i} \ bits$$

155

Average Information Content (Entropy)

- The total information conveyed by the source is

$$I_T = p_1 \times L \times \log_2 \frac{1}{p_1} + p_2 \times L \times \log_2 \frac{1}{p_2} + \cdots p_N \times L \times \log_2 \frac{1}{p_N}$$

- The average information conveyed by the source by emitting L symbols is denoted by its entropy H[x]

$$H[x] = \frac{I_T}{L} = p_1 \times \log_2 \frac{1}{p_1} + p_2 \times \log_2 \frac{1}{p_2} + \cdots p_N \times \log_2 \frac{1}{p_N}$$

156

Entropy

$$\mathrm{H}[x] = -\sum_{x} p(x) \log_2 p(x)$$

Important quantity in

- coding theory
- statistical physics
- machine learning

157

Entropy

Consider a discrete random variable **x** with 8 possible states, each of which is equally likely,

How many bits to transmit the state of **x**?

$$\mathrm{H}[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits}.$$

158

Conditional Entropy

- Suppose we have a joint distribution ***p(x, y)*** from which we draw ***x*** and ***y***.
- If ***x*** is already known, then the additional information needed to specify ***y*** is given by ***–ln p(y|x)***.
- Thus the average additional information needed to specify ***y*** can be written as

$$\mathrm{H}[\mathbf{y}|\mathbf{x}] = -\iint p(\mathbf{y},\mathbf{x}) \ln p(\mathbf{y}|\mathbf{x})\, \mathrm{d}\mathbf{y}\, \mathrm{d}\mathbf{x}$$

$$\mathrm{H}[\mathbf{x},\mathbf{y}] = \mathrm{H}[\mathbf{y}|\mathbf{x}] + \mathrm{H}[\mathbf{x}]$$

WEEK 3 – BAYESIAN LEARNING

Bayesian learning

- Basics (T1 book by Tom Mitchell - 6.1)
- Bayes Theorem (T1 book by Tom Mitchell - 6.2)
- MAP Hypothesis (T1 book by Tom Mitchell - 6.3)
- MLE Hypothesis (T1 book by Tom Mitchell - 6.4)
- Minimum Description Length (MDL) principle

(T1 book by Tom Mitchell - 6.6)

Bayesian Learning

- Bayesian learning algorithms that calculate explicit probabilities for hypotheses, such as the naive Bayes classifier, are among the most practical approaches to certain types of learning problems
- For example: Problem of learning to classify text documents such as electronic news articles.
- For such learning tasks, the naive Bayes classifier is among the most effective algorithms known

161

Features of Bayesian learning

- Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.
- Flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example.
- Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery").

162

Features of Bayesian learning

- Prior knowledge can be combined with observed data to determine the final probability of a hypothesis.
- Prior knowledge is provided by asserting
  - prior probability for each candidate hypothesis, and
  - probability distribution over observed data for each possible hypothesis.
- New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.

Practical Issues

- Require initial knowledge of many probabilities
  - Often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
- Significant computational cost required to determine the Bayes optimal hypothesis in the general case (linear in the number of candidate hypotheses)

Probability Theory

- Classical Definition
  - Consider an experiment which has N equally likely outcomes, and let n of these outcomes correspond to a specific event A, then
  - P(A) = $\frac{\#\ successful\ outcomes}{\#\ possible\ outcomes}$ = n / N

165

Images taken from "Probabilistic Graphical Models" by David Sontag

165

AXIOMS of Probability Theory

- A probability p(ω) for each outcome ω must satisfy the following axioms

$$p(\omega) \geq 0, \quad \sum_{\omega \in \Omega} p(\omega) = 1$$

E.g., p( ) = .6

p( ) = .4

166

Images taken from "Probabilistic Graphical Models" by David Sontag

166

Random Variable

Notation

- A **random variable X** represents outcomes or states of the world.
- We will write p(x) to mean Probability(X = x)
- **Sample space**: the space of all possible outcomes (may be discrete, continuous, or mixed)
- p(x) is the **probability mass (density) function**
  - Assigns a number to each point in sample space
  - Non-negative, sums (integrates) to 1
  - Intuitively: how often does x occur, how much do we believe in x.

167

Probability Distributions

- The outcomes for random variables and their associated probabilities can be organized in to distributions
- Two types of distributions based on types of Random variables: Discrete and Continuous
- Discrete:
  - Binomial, Poisson, Geometric distributions
- Continuous
  - Gaussian, exponential, t, F, chi-squared distibutions

168

Describing distributions

- One way is to construct a graph and analyze the graph to make inferences
  - Discrete: Prob Mass Function (pmf), Cumulative density function
  - Continuous: prob density function (pdf)
- Mean, variance and standard deviations to represent the entire distribution

169

Bernoulli Distribution

- A r.v. X is said to follow Bernoulli's distribution when there are only two possible outcomes
  - By convention either Success (1) or Failure (0)
  - And there is only one trial
  - Ex: tossing a coin at the start f the match
- Let p represents the probability of success and (1-p) represent the probability of failure, then the probability mass function is defined as

$$f(x) = \begin{cases} p & \text{if } x=1 \\ 1-p & x=0 \end{cases}$$

$$p^x (1-p)^{(1-x)}$$

170

Binomial Distribution

- Again binary outcomes, but for n independent trials
  - Probability of success (p) remains the same for all the trials
  - Probability of r success is given by

$P(X=r) = {}^{n}C_{r}\, p^{r}\, (1-p)^{n-r}$

Mean $E(X) = np$

Variance $Var(X) = npq$ (where $q=1-p$)

Estimate Probabilities from Data

- For continuous attributes:
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, use it to estimate the conditional probability $P(X_i|Y)$

Probability Densities

Continuous Probability Distribution

Let X be a continuous rv. Then a *probability distribution or probability density function (pdf)* of X is a function $f(x)$ such that for any two numbers a and b,

$$P(a \le X \le b) = \int_a^b f(x)\,dx$$

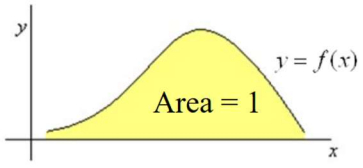
The graph of f is the *density curve*.

Probability Densities

Probability Density Function

For $f(x)$ to be a pdf

1. $f(x) > 0$ for all values of x.
2. The area of the region between the graph of f and the $x-$ axis is equal to 1.

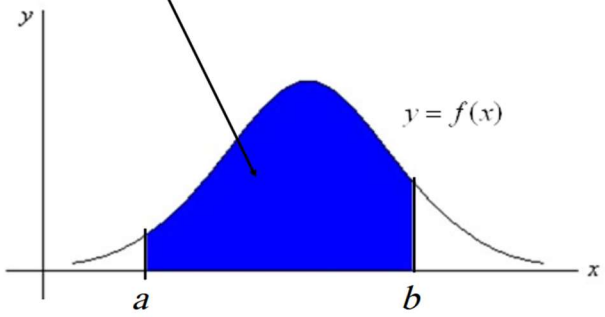


Probability Densities

innovate achieve lead

Probability Density Function

$P(a \leq X \leq b)$ is given by the area of the shaded region.

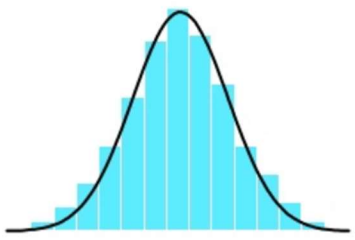


175

Gaussian Distribution

innovate achieve lead

- The commonest and the most useful continuous distribution.
- A symmetrical probability distribution where most results are located in the middle and few are spread on both sides.
- It has the shape of a bell.
- Can entirely be described by its mean and standard deviation.



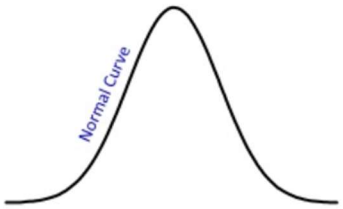
176

Gaussian Distribution

innovate achieve lead

Normal Curve:

- A graphical representation of the normal distribution.
- It is determined by the mean and the standard deviation.
- It a symmetric unimodal bell-shaped curve.
- Its tails extending infinitely in both directions.
- The wider the curve, the larger the standard deviation and the more variation exists in the process.
- The spread of the curve is equivalent to six times the standard deviation of the process.

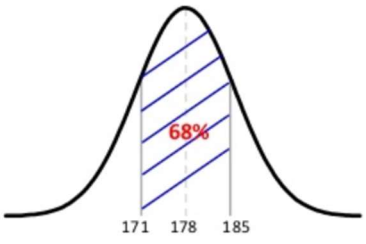


177

Gaussian Distribution

innovate achieve lead

- Suppose that the heights of a sample men are normally distributed.
- The mean height is **178** cm and a standard deviation is **7** cm.
- **We can generalize that:**
  - **68%** of population are between **171** cm and **185** cm.
  - This might be a generalization, but it's true if the data is normally distributed.



178

innovate achieve lead

Mean, Variance & Standard Deviation

- The mean of a discrete random variable is the ***weighted average*** of all of its values. The weights are the probabilities.
- This parameter is also called the expected value of X and is represented by E(X).

$$E(X) = \mu = \sum_{all\ x} xP(x)$$

- The variance is

$$V(X) = \sigma^2 = \sum_{all\ x} (x-\mu)^2 P(x)$$

- The standard deviation is

$$\sigma = \sqrt{\sigma^2}$$

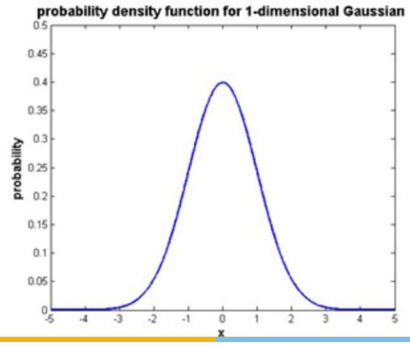
179

innovate achieve lead

Gaussian Distribution

In one dimension

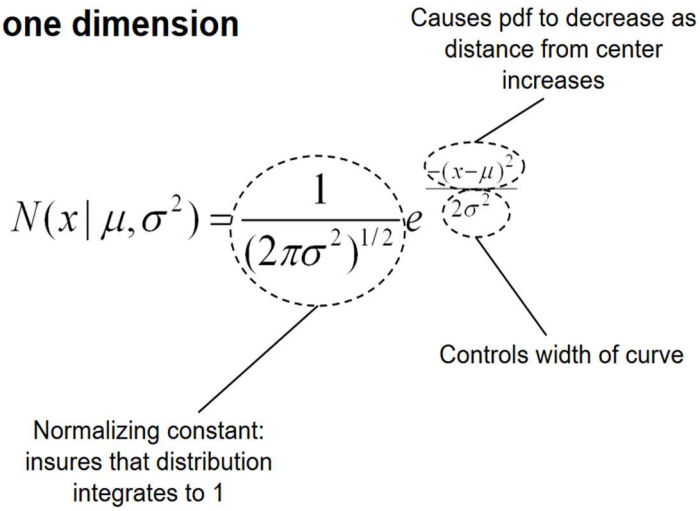
$$N(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$



180

Gaussian Distribution

In one dimension



181

Estimate Probabilities from Data

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(X_i \mid Y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{\frac{(X_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (X_i, Y_i) pair
- For (Income, Class=No):
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{\frac{(120-110)^2}{2(2975)}} = 0.0072$$

182

182

JOINT Distributions

- Probability distribution of two random variables X $\{x_1, x_2, \ldots x_n\}$ and Y$\{y_1, y_2 .. y_k\}$
  - Occurrence of X=xi and Y=yi together
- Example:
  - P(X=0, Y<=1)
  - P(X=1)

    $= \sum_{y=0}^{2} P(X = 1, Y)$

    = 1/6 + 1/6 + 1/8

X \ Y	0	1	2
0	1/4	1/6	1/8
1	1/6	1/6	1/8

183

BITS Pilani, Pilani Campus

183

JOINT Distribution

- Marginal Distribution
  - Sum over any one variable is called Marginal Distribution

P(X=x) = $\sum_{y \in Y} P(X, Y)$

P(Y=y) = $\sum_{x \in X} P(X, Y)$

184

BITS Pilani, Pilani Campus

184

Conditional Probability

- Estimating probability for two or more related events
  - Measure influence of one variable over another
- Let A and B be two events, p(B) > 0

  $p(A|B) = p(A \cap B) / p(B)$
- Using random variable notations,
  - p(a|b) denotes the probability of A=a and B=b
    - i.e. p(A=a |B=b)

185

185

Basic Formulas for Probabilities

- *Product Rule*: probability $P(A \wedge B)$ of a conjunction of two events A and B:

  $$P(A \wedge B) = P(A \mid B)\, P(B) = P(B \mid A)\, P(A)$$
- *Sum Rule*: probability of a disjunction of two events A and B:

  $$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$
- *Theorem of total probability*: if events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then

  $$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

186

186

Example 1

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B \mid A)P(A) + P(B \mid \sim A)P(\sim A)}$$

A = you have the flu, B = you just coughed Assume:

P(A) = 0.05
P(B|A) = 0.80
P(B| ~A) = 0.20

what is P(flu | cough) = P(A|B)?

187

Example 2

- Does a patient have cancer or not?
  - The patient takes a lab test and the test returns a correct positive result in only 98% of the cases in which the disease is actually present
  - And a correct negative result in only 97% of the cases in which the disease is not present
  - Of the total population, only 0.008% has cancer

188

Example 2

Given:

$P(cancer) =$	$P(\neg cancer) =$
$P(+\mid cancer) =$	$P(-\mid cancer) =$
$P(+\mid \neg cancer) =$	$P(-\mid \neg cancer) =$

Using Bayes Theorem:

$P(\text{cancer}\mid +) = P(+\mid cancer)\, P(cancer) / P(+)$

$P(\neg cancer \mid +) = P(+\mid \neg cancer)\, P(\neg cancer) / P(+)$

Since denominator is common

$P(\text{cancer}\mid +)$ *proportional to* $P(+\mid cancer)\, P(cancer)$

$P(\neg cancer \mid +)$ *proportional to* $P(+\mid \neg cancer)\, P(\neg cancer)$

189

Example 2

$P(cancer) = 0.008$	$P(\neg cancer) = 1-0.008=0.992$
$P(+\mid cancer) = 0.98$	$P(-\mid cancer) = 0.02$
$P(+\mid \neg cancer) = 1-0.97=0.3$	$P(-\mid \neg cancer) = 0.97$

190

innovate achieve lead

Remember: Some terminology

- Likelihood function: P(data | θ)
- Prior: P(θ)
- Posterior: P(θ | data)

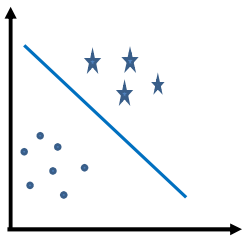
- **Conjugate prior**: P(θ) is the conjugate prior for likelihood function P(data | θ) if the forms of P(θ) and P(θ | data) are the same.

191

innovate achieve lead

Machine Learning

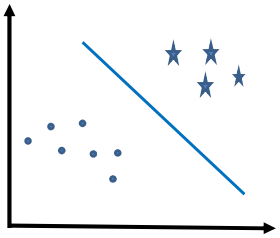
- Objective of ML: Given data X and target variable Y, determine the joint distribution P(X, Y)
- Classification Problem
  - Decision boundary that separates one class from another class
  - Determine P(Y|X)
  - These models are called Deterministic



192

Machine Learning

- Alternate approach is to understand the process that generated the data
  - Generative Models P(X,Y)
  - Build a model for all the positive cases or category 1
  - Build another model for all the negative cases or category 2
  - For predicting a new test case
    - Run the test case with both the models and choose the model with maximum probability



193

Machine Learning

- Generative models
  - Build model to estimate the posterior probability P(Y|X) by estimating
  - likelihood of data given target (hypothesis) P(X|Y)
  - Prior probabilities over target P(Y)
  - In general, for a specific class Y=ck,

$$P(Y = c_k|X) = \frac{P(X|Y = c_k) * P(Y=c_k)}{P(X)}$$

194

Hypothesis

- Relationship between the input and output values.
- Lets say that **target function** $y=f(\mathbf{x})$ However, f(.) is unknown function to us.
- Machine learning algorithms try to guess a hypothesis function $h(\mathbf{x})$ that approximates the unknown $f(.)$
- Set of all possible hypotheses is known as the Hypothesis set or space H(.)
- Goal is the learning process is to find the final hypothesis that best approximates the unknown target function.

195

Bayes Theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- $P(h)$ = prior probability of hypothesis h
- $P(D)$ = prior probability of training data D
- $P(h|D)$ = probability of h given D
- $P(D|h)$ = probability of D given h

196

196

Choosing Hypotheses

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- Generally want the most probable hypothesis given the training data
  Maximum a posteriori hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h|D) \\ &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D|h)P(h) \end{aligned}$$

- If assume $P(h_i) = P(h_j)$ then can further simplify, and choose the *Maximum likelihood* (ML) hypothesis

$$h_{ML} = \arg\max_{h_i \in H} P(D|h_i)$$

197

197

Brute Force MAP Hypothesis

1. For each hypothesis h in H, calculate the posterior probability

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

1. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \operatorname*{argmax}_{h \in H} P(h|D)$$

198

198

Practical Issues

- Require initial knowledge of many probabilities
  - Often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
- Significant computational cost required to determine the Bayes optimal hypothesis in the general case (linear in the number of candidate hypotheses)

199

MAP Hypothesis

- Using Bayes theorem, we compute the MAP hypothesis for all probable hypothesis (or all unique class labels)
- Identify the best hypothesis describing the data as

$$h_{MAP} = argmax_{h \in H} P(h|D)$$

$$= argmax_{h \in H} \frac{P(D|h)P(h)}{P(D)}$$

$$= argmax_{h \in H} P(D|h)P(h)$$

H: set of all hypothesis

P(D) is independent of h and is same for all hypothesis, therefore dropped

200

Maximum Likelihood estimation

- When no prior information is available, all hypothesis are equally likely i.e. p(hi) = p(hj)
  - This is also true for a balanced class problem where all the classes are equally likely
  - This is known as Uniform prior
  - MAP hypothesis further simplifies to:

$$H_{ML} = argmax_{h \in H} \mathrm{P}(\mathrm{D}|\mathrm{h})$$

This is called Maximum Likelihood Hypothesis

201

ML setting

- Bayesian Analysis
  - start with some belief about the system, called a prior.
  - Then we obtain some data and use it to update our belief.
  - The outcome is called a posterior.
  - Should we obtain even more data, the old posterior becomes a new prior and the cycle repeats.
  - People often use likelihood for evaluation of models: a model that gives higher likelihood to real data is better

202

ML Setting

- P(h | D) a posterior determines the class label
- It's a probability distribution over model parameters obtained from prior beliefs and data.
- When one uses likelihood to get point estimates of model parameters, it's called Maximum Likelihood estimation or MLE.
- If one also takes the prior into account, then it's maximum a posteriori estimation (MAP).
- MLE and MAP are the same if the prior is uniform
- This forms the basis for Naïve Bayes classifier

Relation to Concept Learning

- Consider our usual concept learning task
  - instance space *X*, hypothesis space *H*, training examples *D*
  - consider the FindS learning algorithm (outputs most specific hypothesis from the version space $VS_{H,D}$)
- What would Bayes rule produce as the MAP hypothesis?
- Does *FindS* output a MAP hypothesis??

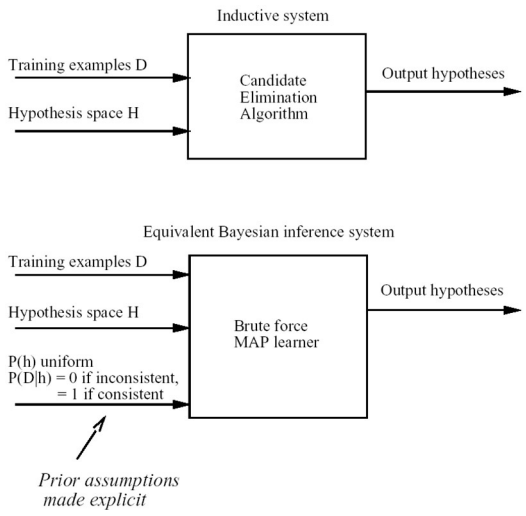
Relation to Concept Learning

- Assume fixed set of instances $\langle x_1, \ldots, x_m \rangle$
- Assume D is the set of classifications: $D = \langle c(x_1), \ldots, c(x_m) \rangle$
- Choose $P(D|h)$:
  - $P(D|h) = 1$ if h consistent with D
  - $P(D|h) = 0$ otherwise
- Choose $P(h)$ to be *uniform* distribution
  - $P(h) = 1/|H|$ for all h in H
- Then,

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}$$

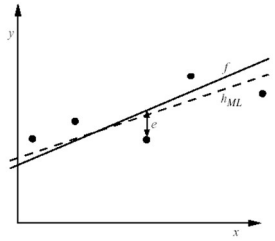
205

Characterizing Learning Algorithms by Equivalent MAP Learners



206

Learning A Real Valued Function



Consider any real-valued target function *f*

Training examples $\langle x_i, d_i \rangle$, where d_i is noisy training value

- $d_i = f(x_i) + e_i$
- e_i is random variable (noise) drawn independently for each x_i according to some Gaussian distribution with mean=0

Then the maximum likelihood hypothesis h_{ML} is the one that minimizes the sum of squared errors: $h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$

Learning A Real Valued Function

$$\begin{aligned} h_{ML} &= \underset{h \in H}{\operatorname{argmax}} p(D|h) \\ &= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} p(d_i|h) \\ &= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2} \end{aligned}$$

- Maximize natural log of this instead...

$$\begin{aligned} h_{ML} &= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\ &= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\ &= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -(d_i - h(x_i))^2 \\ &= \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2 \end{aligned}$$

Learning to Predict Probabilities

- Consider predicting survival probability from patient data
- Training examples $\langle x_i, d_i \rangle$, where d_i is 1 or 0
- Want to train neural network to output a *probability* given x_i (not a 0 or 1)
- In this case can show

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

- Weight update rule for a sigmoid unit:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

where

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} (d_i - h(x_i))\ x_{ijk}$$

Minimum Description Length Principle

Occam's razor: prefer the shortest hypothesis
MDL: prefer the hypothesis *h* that minimizes

$$h_{MDL} = \underset{h \in H}{\operatorname{argmin}} L_{C_1}(h) + L_{C_2}(D|h)$$

where $L_C(x)$ is the description length of x under encoding C

Example: H = decision trees, D = training data labels

- $L_{C1}(h)$ is # bits to describe tree h
- $L_{C2}(D|h)$ is # bits to describe D given h
  - Note $L_{C2}(D|h) = 0$ if examples classified perfectly by h. Need only describe exceptions
- Hence h_{MDL} trades off tree size for training errors

Minimum Description Length Principle

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$
$$= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h)$$
$$= \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \quad (1)$$

Interesting fact from information theory:

The optimal (shortest expected coding length) code for an event with

probability p is $-\log_2 p$ bits.

So interpret (1):

- $-\log_2 P(h)$ is length of h under optimal code
- $-\log_2 P(D|h)$ is length of D given h under optimal code

→ prefer the hypothesis that minimizes

length(*h*) + *length*(*misclassifications*)

211

BITS Pilani, Pilani Campus

211

Most Probable Classification of New Instances

- So far we've sought the most probable *hypothesis* given the data D (i.e., h_{MAP})
- Given new instance x, what is its most probable *classification*?
  - $h_{MAP}(x)$ is not the most probable classification!
- Consider:
  - Three possible hypotheses:

    $P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$
  - Given new instance x,

    $h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$
  - What's most probable classification of x?

212

BITS Pilani, Pilani Campus

212

Bayes Optimal Classifier

- **Bayes optimal classification:**

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- Example:

$P(h_1|D) = .4,\ P(-|h_1) = 0,\ P(+|h_1) = 1$
$P(h_2|D) = .3,\ P(-|h_2) = 1,\ P(+|h_2) = 0$
$P(h_3|D) = .3,\ P(-|h_3) = 1,\ P(+|h_3) = 0$

therefore

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

and

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

Gibbs Classifier

- Bayes optimal classifier provides best result, but can be expensive if many hypotheses.
- Gibbs algorithm:
  1. Choose one hypothesis at random, according to $P(h|D)$
  2. Use this to classify new instance
- Surprising fact: Assume target concepts are drawn at random from H according to priors on H. Then:

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptional}]$$

- Suppose correct, uniform prior distribution over H, then
  - Pick any hypothesis from VS, with uniform probability
  - Its expected error no worse than twice Bayes optimal

WEEK 4 - Probabilistic Generative Classifiers

- Bayesian learning algorithms that calculate explicit probabilities for hypotheses, such as the naive Bayes classifier, are among the most practical approaches to certain types of learning problems
- For example: Problem of learning to classify text documents such as electronic news articles.
- For such learning tasks, the naive Bayes classifier is among the most effective algorithms known

215

Features of Bayesian learning

- Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.
- Flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example.
- Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery").

216

Features of Bayesian learning

innovate achieve lead

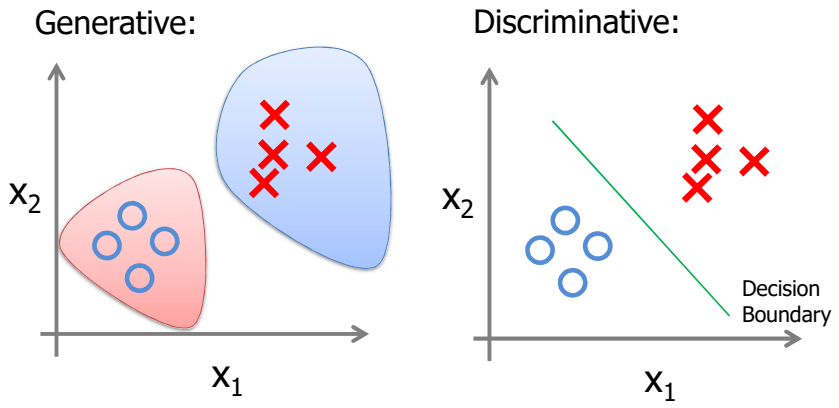
- Prior knowledge can be combined with observed data to determine the final probability of a hypothesis.
- Prior knowledge is provided by asserting
  - prior probability for each candidate hypothesis, and
  - probability distribution over observed data for each possible hypothesis.
- New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.

Probabilistic Generative Classifiers

innovate achieve lead

- Approach is to understand the process that generated the data
  - Generative Models P(X,Y)
  - Build a model for all the positive cases or category 1
  - Build another model for all the negative cases or category 2
  - For predicting a new test case
    - Run the test case with both the models and choose the model with maximum probability

Probabilistic Generative Model versus Probabilistic Discriminative Model



219

Probabilistic Models: Generative/Discriminative

- Model $p(C_k|\boldsymbol{x})$ in an *inference* stage and use it to make optimal decisions
- Approaches to computing the $p(C_k|\boldsymbol{x})$
  1. Generative
     - Model class conditional densities by $p(\boldsymbol{x}|C_k)$ together with prior probabilities $p(C_k)$
     - Then use Bayes rule to compute posterior

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{p(\boldsymbol{x})}$$

  2. Discriminative
     - Directly model conditional probabilities $p(C_k|\boldsymbol{x})$

13

220

220

Probabilistic Generative Model versus Probabilistic Discriminative Model

Generative	Discriminative
Ex: Naïve Bayes	Ex: Logistic Regression
Estimate $P(Y)$ and $P(X\|Y)$	Finds class label directly $P(Y\|X)$
Prediction $\hat{y} = \text{argmax}_y\, P(Y = y)P(X = x\|Y = y)$	Prediction $\hat{y} = P(Y = y\|X = x)$

Generative Models

- Generative models
  - Build model to estimate the posterior probability P(Y|X) by estimating
  - likelihood of data given target (hypothesis) P(X|Y)
  - Prior probabilities over target P(Y)
  - In general, for a specific class Y=ck,

$$P(Y = c_k|X) = \frac{P(X|Y = c_k) * P(Y=c_k)}{P(X)}$$

Most Probable Classification of New Instances

- So far we've sought the most probable *hypothesis* given the data D (i.e., h_{MAP})
- Given new instance x, what is its most probable *classification*?
  - $h_{MAP}(x)$ is not the most probable classification!
- Consider:
  - Three possible hypotheses:

    $P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$
  - Given new instance x,

    $h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$
  - What's most probable classification of x?

223

Bayes Optimal Classifier

- **Bayes optimal classification:**

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- Example:

$P(h_1|D) = .4,\ P(-|h_1) = 0,\ P(+|h_1) = 1$
$P(h_2|D) = .3,\ P(-|h_2) = 1,\ P(+|h_2) = 0$
$P(h_3|D) = .3,\ P(-|h_3) = 1,\ P(+|h_3) = 0$

therefore

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

and

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

224

Gibbs Classifier

- Bayes optimal classifier provides best result, but can be expensive if many hypotheses.
- Gibbs algorithm:
  1. Choose one hypothesis at random, according to $P(h|D)$
  2. Use this to classify new instance
- Surprising fact: Assume target concepts are drawn at random from H according to priors on H. Then:

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptional}]$$

- Suppose correct, uniform prior distribution over H, then
  - Pick any hypothesis from VS, with uniform probability
  - Its expected error no worse than twice Bayes optimal

225

225

Conditional independence

- **Definition**: X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k)\ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z_k)$$

$$P(X|Y, Z) = P(X|Z)$$

Example:

$$P(\text{Thunder}|\text{Rain}, \text{Lightning}) = P(\text{Thunder}|\text{Lightning})$$

Slide credit: Tom Mitchell

226

Applying conditional independence

- Naïve Bayes assumes X_i are conditionally independent given Y

e.g., $P(X_1|X_2, Y) = P(X_1|Y)$

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

General form: $P(X_1, \cdots, X_n|Y) = \prod_{j=1}^{n} P(X_j|Y)$

How many parameters to describe $P(X_1, \cdots, X_n|Y)$? $P(Y)$?

- Without conditional independence assumption?
- With conditional independence assumption?

Slide credit: Tom Mitchell

227

Naïve Bayes Independence assumption

- Assumption:

$$P(X_1, \cdots, X_n|Y) = \prod_{j=1}^{n} P(X_j|Y)$$

- i.e., X_i and X_j are conditionally independent given Y for $i \neq j$

Slide credit: Tom Mitchell

228

Naïve Bayes classifier

innovate achieve lead

- Bayes rule:

$$P(Y = y_k | X_1, \cdots, X_n) = \frac{P(Y = y_k) P(X_1, \cdots, X_n | Y = y_k)}{\sum_j P(Y = y_j) P(X_1, \cdots, X_n | Y = y_j)}$$

- Assume conditional independence among X_i's:

$$P(Y = y_k | X_1, \cdots, X_n) = \frac{P(Y = y_k) \Pi_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \Pi_i P(X_i | Y = y_j)}$$

- Pick the most probable (MAP) Y

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}} P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

Prior Probability

MLE

Slide credit: Tom Mitchell

NAÏVE BAYES CLASSIFIER

innovate achieve lead

- Assume independence among attributes X_i when class is given:
  - $P(X_1, X_2, \ldots, X_d | Y_j) = P(X_1 | Y_j) P(X_2 | Y_j) \ldots P(X_d | Y_j)$
  - Now we can estimate $P(X_i | Y_j)$ for all X_i and Y_j combinations from the training data
  - New point is classified to Y_j if $P(Y_j) \prod P(X_i | Y_j)$ is maximal.

230

Slide adopted from "Introduction to Data mining" Vipin Kumar

Example 1

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

A: attributes

M: mammals

N: non-mammals

$$P(A \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

231

231

Naive Bayes Classifier

- Assume target function $f : X \rightarrow V$, where each instance x described by attributes $<a_1, a_2 \ldots a_n>$.
- Most probable value of $f(x)$ is:

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n)$$

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

Naive Bayes assumption:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

Naive Bayes classifier:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

232

232

Naive Bayes Algorithm

- Naive Bayes Learn(*examples*)

  For each target value v_j

  $$\hat{P}(v_j) \leftarrow \text{estimate } P(v_j)$$

  For each attribute value a_i of each attribute a

  $$\hat{P}(a_i \mid v_j) \leftarrow \text{estimate } P(a_i \mid v_j)$$

- Classify New Instance(*x*)

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i | v_j)$$

233

BITS Pilani, Pilani Campus

233

Naive Bayes: Example

- Consider *PlayTennis*, and new instance

  <Outlk = sun, Temp = cool, Humid = high, Wind = strong>

- Want to compute:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

$P(y)$ P($sun|y$) $P(cool|y)$ $P(high|y)$ P($strong|y$) = .005

$P(n)$ P($sun|n$) $P(cool|n)$ $P(high|n)$ P($strong|n$) = .021

$\rightarrow v_{NB} = n$

234

BITS Pilani, Pilani Campus

234

Issues with Naïve Bayes Classifier

Naïve Bayes Classifier:

P(Refund = Yes | No) = 3/7
P(Refund = No | No) = 4/7
P(Refund = Yes | Yes) = 0
P(Refund = No | Yes) = 1
P(Marital Status = Single | No) = 2/7
P(Marital Status = Divorced | No) = 1/7
P(Marital Status = Married | No) = 4/7
P(Marital Status = Single | Yes) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
P(Marital Status = Married | Yes) = 0

For Taxable Income:
If class = No: sample mean = 110
sample variance = 2975
If class = Yes: sample mean = 90
sample variance = 25

- P(Yes) = 3/10
  P(No) = 7/10
- P(Yes | Married) = 0 x 3/10 / P(Married)
  P(No | Married) = 4/7 x 7/10 / P(Married)

Issues with Naïve Bayes Classifier

Consider the table with Tid = 7 deleted

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier:

P(Refund = Yes | No) = 2/6
P(Refund = No | No) = 4/6
P(Refund = Yes | Yes) = 0
P(Refund = No | Yes) = 1
P(Marital Status = Single | No) = 2/6
P(Marital Status = Divorced | No) = 0
P(Marital Status = Married | No) = 4/6
P(Marital Status = Single | Yes) = 2/3
P(Marital Status = Divorced | Yes) = 1/3
P(Marital Status = Married | Yes) = 0/3
For Taxable Income:
If class = No: sample mean = 91
sample variance = 685
If class = No: sample mean = 90
sample variance = 25

Given X = (Refund = Yes, Divorced, 120K)
P(X | No) = 2/6 X 0 X 0.0083 = 0
P(X | Yes) = 0 X 1/3 X 1.2 X 10^{-9} = 0

Naïve Bayes will not be able to classify X as Yes or No!

Issues with Naïve Bayes Classifier

- If one of the conditional probabilities is zero, then the entire expression becomes zero
- Need to use other estimates of conditional probabilities than simple fractions
- Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m-estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability of the class

m: parameter

N_c: number of instances in the class

N_{ic}: number of instances having attribute value A_i in class c

237

Slide adopted from "Introduction to Data mining" Vipin Kumar

237

A Simple Example

Text	Tag
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Which tag does the sentence *A very close game* belong to? i.e. P(sports| *A very close game*)

Feature Engineering: Bag of words i.e use word frequencies without considering order

Using Bayes Theorem:

P(sports| ***A very close game*****)**

$$= \frac{\textbf{P(}\textit{A very close game}\textbf{|}\ \textbf{sports) P(sports)}}{\textbf{P(}\textit{A very close game}\textbf{)}}$$

We assume that every word in a sentence is **independent** of the other ones

$$P(a\ very\ close\ game) = P(a) \times P(very) \times P(close) \times P(game)$$

$$P(a\ very\ close\ game|Sports) = P(a|Sports) \times P(very|Sports) \times P(close|Sports) \times P(game|Sports)$$

"close" doesn't appear in sentences of sports tag, So P(close | sports) = 0, which makes product 0

238

238

Laplace smoothing

- Laplace smoothing: we add 1 or in general constant k to every count so it's never zero.
- To balance this, we add the number of possible words to the divisor, so the division will never be greater than 1
- In our case, the possible words are ['a', 'great', 'very', 'over', 'it', 'but', 'game', 'election', 'clean', 'close', 'the', 'was', 'forgettable', 'match'].

239

Apply Laplace Smoothing

Word	P(word | Sports)	P(word | Not Sports)
a	2+1 / 11+14	1+1 / 9+14
very	1+1 / 11+14	0+1 / 9+14
close	0+1 / 11+14	1+1 / 9+14
game	2+1 / 11+14	0+1 / 9+14

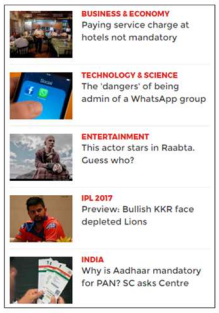
$$P(a|Sports) \times P(very|Sports) \times P(close|Sports) \times P(game|Sports) \times P(Sports)$$
$$= 2.76 \times 10^{-5}$$
$$= 0.0000276$$

$$P(a|Not\ Sports) \times P(very|Not\ Sports) \times P(close|Not\ Sports) \times P(game|Not\ Sports) \times P(Not\ Sports)$$
$$= 0.572 \times 10^{-5}$$
$$= 0.00000572$$

240

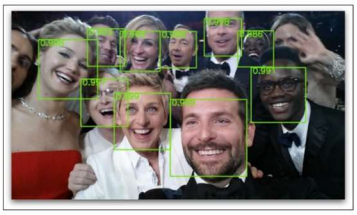
Naïve Bayes Classifier Applications

Categorizing News



Email Spam Detection

Face Recognition



Sentiment Analysis

241

Naive Bayes Classifier

- Along with decision trees, neural networks, one of the most practical learning methods.
- When to use
  - Moderate or large training set available
  - Attributes that describe instances are conditionally independent given classification
- Successful applications:
  - Diagnosis
  - Classifying text documents

242

Learning to Classify Text

- Why?
  - Learn which news articles are of interest
  - Learn to classify web pages by topic
- Naive Bayes is among most effective algorithms
- What attributes shall we use to represent text documents??

Learning to Classify Text (2/4)

Target concept Interesting? : *Document* →{+, –}

1. Represent each document by vector of words
- one attribute per word position in document

2. Learning: Use training examples to estimate
- $P(+)$
- $P(-)$
- $P(doc|+)$
- $P(doc|-)$

Naive Bayes conditional independence assumption

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

where $P(a_i = w_k \mid v_j)$ is probability that word in position i is w_k, given v_j

one more assumption: $P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

Learning to Classify Text (3/4)

LEARN_NAIVE_BAYES_TEXT (*Examples*, V)

1. *collect all words and other tokens that occur in Examples*

- *Vocabulary* ← all distinct words and other tokens in *Examples*

2. *calculate the required* $P(v_j)$ *and* $P(w_k \mid v_j)$ *probability terms*

- For each target value v_j in *V* do
  - $docs_j \leftarrow$ subset of *Examples* for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$

245

BITS Pilani, Pilani Campus

245

Learning to Classify Text (4/4)

  - $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)
  - for each word w_k in *Vocabulary*
    - $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
    - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

CLASSIFY_NAIVE_BAYES_TEXT (*Doc*)

- *positions* ← all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return v_{NB} where $v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_{i \in positions} P(a_i|v_j)$

246

BITS Pilani, Pilani Campus

246

innovate achieve lead

Baseline: Bag of Words Approach

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

247

innovate achieve lead

Case Study: Text Classification

- Classify e-mails
  - Y = {Spam, NotSpam}
- Classify news articles
  - Y = what is the topic of the article?
- Classify webpages
  - Y = {student, professor, project, …}
- What about the features X?
  - The text!

248

innovate achieve lead

Features X are entire document - X_i for ith word in article

Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.e
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinic
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided
```


249

innovate achieve lead

Naïve Bayes for Text Classification

- **Naïve Bayes assumption helps a lot!**
  - $P(X_i = x_i|Y = y)$ is just the probability of observing word x_i at the ith position in a document on topic y.
  - Assume X_i is independent of all other words in document given the label y: $P(X_i = x_i|Y = y, X_{-i}) = P(X_i = x_i|Y = y)$.

$$h_{NB}(x) = \arg\max_y P(y) \prod_{i=1}^{\text{lengthDoc}} P(X_i = x_i|y)$$

  - For each label y, have 1000 distributions of size 10000 to estimate.
  - This is 10000×1000 items, which is big but much less than 10000^{1000}...

250

innovate achieve lead

Bag of Words Model

- Typical additional assumption – **Position in document doesn't matter**:

$$P(X_i = x_i \mid Y = y) = P(X_k = x_i \mid Y = y)$$

the probability distributions of words are the same at each position: $P_i = P_j$ for all i, j.

- **"Bag of Words"** model – order of words in the document is ignored
- Now, only 10000 quantities $P(x_i|y)$ to estimate for each label y (the 10000 possible values that x_i can be) plus the prior.

$$h_{NB}(x) = \arg\max_y P(y) \prod_{i=1}^{1000} P(x_i|y)$$

251

innovate achieve lead

Bag of Words model

- Typical additional assumption – **Position in document doesn't matter**:

$$P(X_i = x_i \mid Y = y) = P(X_k = x_i \mid Y = y)$$

- **"Bag of Words"** model – order of words on the page ignored
- Sounds silly but often works very well

A piece of text like "When the lecture is over, remember to take your bag" would look to this algorithm the same as if we just sorted the words alphabetically *"bag is lecture over remember take the to When your"*

252

innovate achieve lead

Bag of Words model

- Typical additional assumption – **Position in document doesn't matter**:

$$P(X_i = x_i \mid Y = y) = P(X_k = x_i \mid Y = y)$$

- **"Bag of Words"** model – order of words on the page ignored

Can simplify further:

$$\prod_{i=1}^{\text{lengthDoc}} P(x_i|y) = \prod_{w=1}^{W} P(w|y)^{\text{count}(w)}$$

253

innovate achieve lead

Bag of Words representation

- Since we are assuming the order of words doesn't matter, an alternative representation of document is as vector of counts:
  - $x^{(j)}$ = number of occurrences of word j in document x.
  - Typical document: [0 0 1 0 0 3 0 0 0 1 0 0 0 1 0 0 2 0 0 ...]
  - Called "bag of words" or "term vector" or "vector space model" representation

254

Naïve Bayes with Bag of Words for text classification

- Learning phase
  - Class Prior $P(Y)$
  - $P(X_i|Y)$
- Test phase:
  - For each document
    - Use naïve Bayes decision rule

$$h_{NB}(x) = \arg\max_y P(y) \prod_{i=1}^{1000} P(x_i|y)$$

255

Twenty NewsGroups

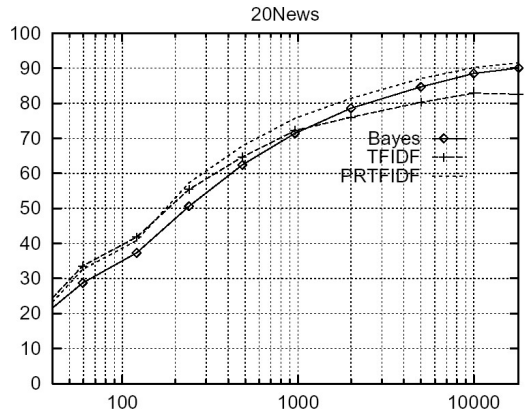
- Given 1000 training documents from each group Learn to classify new documents according to which newsgroup it came from

comp.graphics	misc.forsale	alt.atheism	sci.space
comp.os.ms-windows.misc	rec.autos	soc.religion.christian	sci.crypt
comp.sys.ibm.pc.hardware	rec.motorcycles	talk.religion.misc	sci.electronics
comp.sys.mac.hardware	rec.sport.baseball	talk.politics.mideast	sci.med
comp.windows.x	rec.sport.hockey	talk.politics.misc	
		talk.politics.guns	

- Naive Bayes: 89% classification accuracy

256

Learning Curve for 20 Newsgroups



- Accuracy vs. Training set size (1/3 withheld for test)

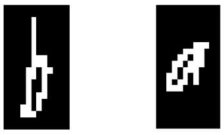
257

257

Estimating Parameters: X_i Continuous

What if features are continuous?

- E.g., character recognition: X_i is intensity at ith pixel



- Gaussian Naïve Bayes (GNB):

$$P(X_i = x | Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

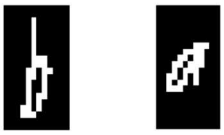
distribution of feature X_i is Gaussian with a mean and variance that can depend on the label y_k and which feature X_i it is

258

innovate achieve lead

What if features are continuous?

- E.g., character recognition: X_i is intensity at ith pixel



- Gaussian Naïve Bayes (GNB):

$$P(X_i = x|Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

- Different mean and variance for each class k and each pixel i.
- Sometimes assume variance:
  - Is independent of Y (i.e., just have σ_i)
  - Or independent of X (i.e., just have σ_k)
  - Or both (i.e., just have σ)

259

innovate achieve lead

Estimating parameters: Y discrete, X_i continuous

- Maximum likelihood estimates:

$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_{j=1}^{N} x_j$$

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

X_i^j: ith pixel in jth training image; $\delta(Y^j = y_k)$: jth training image; y_k: kth class

$$\hat{\sigma}^2_{unbiased} = \frac{1}{N-1}\sum_{j=1}^{N}(x_j - \hat{\mu})^2$$

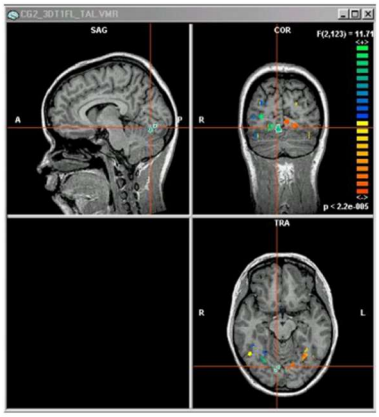
$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k) - 1} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

260

innovate achieve lead

Example: GNB for classifying mental states

[Mitchell et al.]



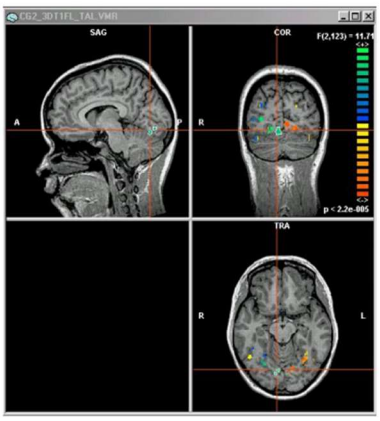
- Classify a person's cognitive state, based on brain image
  - reading a sentence or viewing a picture?
  - reading the word describing a "Tool" or "Building"?
  - reading the word describing a "Person" or an "Animal"?
- Training: Patients were shown words of different categories and then a measurement was done to see what parts of the brain responded.

261

innovate achieve lead

Example: GNB for classifying mental states

[Mitchell et al.]



~1mm resolution

~2 images per sec.

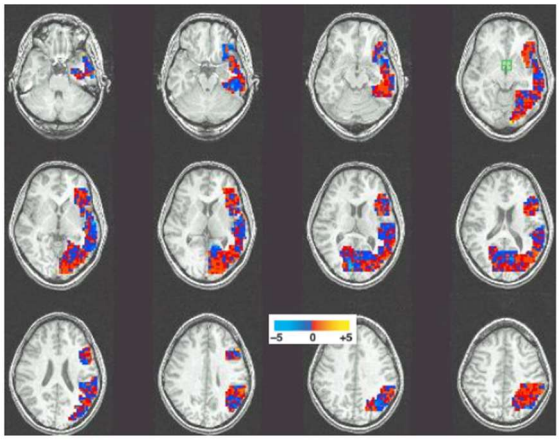
15,000 voxels/image

Non-invasive, save

Measures Blood Oxygen Level Dependent response (BOLD)

262

Gaussian Naïve Bayes: Learned $\mu_{voxel,word}$



[Mitchell et al.]

15,000 voxels or features

10 training examples or subjects per class

263

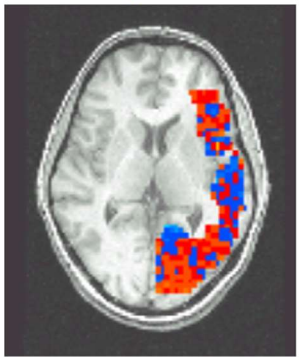
Learned Naïve Bayes Models – Means for P(BrainActivity | WordCategory)

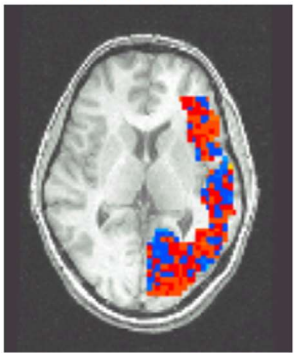
Pairwise classification accuracy: 85%

[Mitchell et al.]

People words

Animal words





264

Practical Issues of Bayesian learning

- Require initial knowledge of many probabilities
  - Often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
- Significant computational cost required to determine the Bayes optimal hypothesis in the general case (linear in the number of candidate hypotheses)

265

WEEK 5 - Probabilistic Discriminative Classifiers

- 4.1 Discriminant Functions,(4.1.1, 4.1.2)
- 4.3 Probabilistic Discriminative Classifiers,
- 4.3.1, 4.3.2 Logistic regression
- Difference between Naïve Bayes Classifier and Logistic Regression

266

Logistic Regression

Idea:

- Naïve Bayes allows computing P(Y|X) by learning P(Y) and P(X|Y)
- Why not learn P(Y|X) directly?

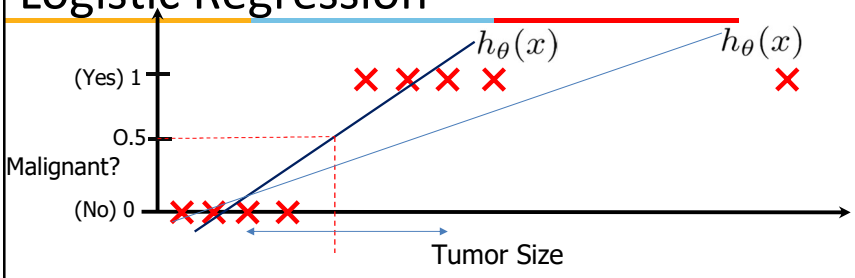
267

Linear Regression versus logistic regression

- **Linear Regression** could help us predict the student's test score on a scale of 0 - 100. Linear regression predictions are continuous (numbers in a range).
- **Logistic Regression** could help use predict whether the student passed or failed. Logistic regression predictions are discrete (only specific values or categories are allowed). We can also view probability scores underlying the model's classifications.

268

Linear Regression versus Logistic Regression



Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

269

Linear Regression versus Logistic Regression

Classification requires discrete values:
y = 0 or 1

For linear Regression output values:

$h_\theta(x)$ can be much > 1 or much < 0

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

270

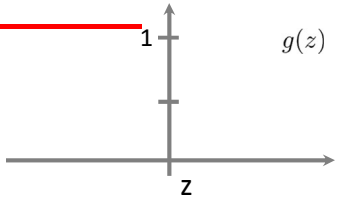
Sigmoid/Logistic Function

- Sigmoid/logistic function takes a real value as input and outputs another value between 0 and 1
- That framework is called logistic regression
  - Logistic: A special mathematical sigmoid function it uses
  - Regression: Combines a weight vector with observations to create an answer

$$h_\theta(x) = g(\theta^T x)$$

Logistic Regression

$$h_\theta(x) = g(\theta^T x)$$
$$g(z) = \frac{1}{1+e^{-z}}$$



Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

predict "$y = 0$" if $h_\theta(x) < 0.5$

Interpretation of Logistic Regression hypothesis

$h_\theta(x)$ = estimated probability that y = 1 on input x

Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$

$h_\theta(x) = 0.7$

Tell patient that 70% chance of tumor being malignant

"probability that y = 1, given x, parameterized by θ"

$$P(y = 0|x;\theta) + P(y = 1|x;\theta) = 1$$
$$P(y = 0|x;\theta) = 1 - P(y = 1|x;\theta)$$

Learning model parameters

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples $\quad x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \quad x_0 = 1, y \in \{0, 1\}$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

How to choose parameters (feature weights) θ ?

Error (Cost) Function

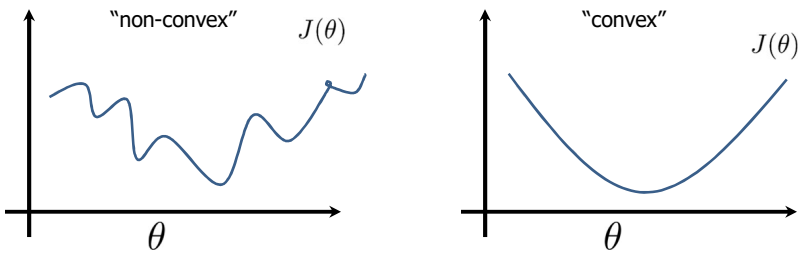
- Our prediction function is non-linear (due to sigmoid transform)
- Squaring this prediction as we do in MSE results in a non-convex function with many local minima.
- If our cost function has many local minimums, gradient descent may not find the optimal global minimum.
- So instead of Mean Squared Error, we use a error/ cost function called Cross-Entropy, also known as Log Loss.

275

MSE Cost Function

Linear regression: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

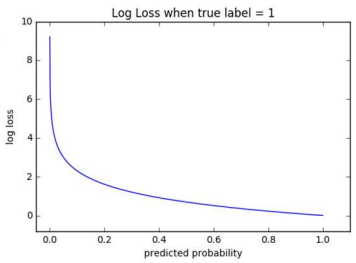
$$\text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$



276

Cross Entropy

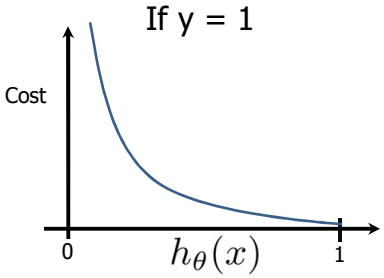
- Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1.
- Cross-entropy loss increases as the predicted probability diverges from the actual label.
- So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value.
- A perfect model would have a log loss of 0.
- Cross-entropy loss can be divided into two separate cost functions: one for y=1 and one for y=0.



277

Logistic regression cost function (cross entropy)

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



$\text{Cost} = 0$ if $y = 1, h_\theta(x) = 1$
But as $h_\theta(x) \to 0$
$Cost \to \infty$

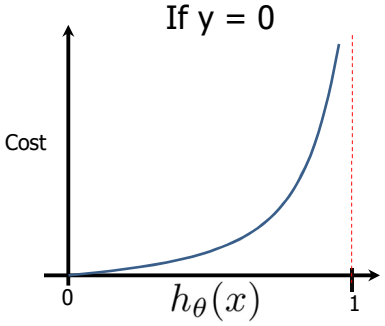
Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

278

Logistic regression cost function

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

If y = 0



Cost=0; If y=0 and $h_\theta(x)=0$

279

Cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

[redacted]

To fit parameters θ : Apply Gradient Descent Algorithm

$$\min_\theta J(\theta)$$

To make a prediction given new x :

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

280

Derivative of sigmoid function

- Maximum likelihood to determine the parameters of the logistic regression model.
- To do this, we shall make use of the derivative of the logistic sigmoid function
- Use any algorithm like the gradient descent algorithm to minimize cost function by using derivative

https://towardsdatascience.com/derivative-of-the-sigmoid-function-536880cf918e

281

How does logistic regression handle missing values?

- Replace missing values with column averages (i.e. replace missing values in feature 1 with the average for feature 1).
- Replace missing values with column medians.
- Impute missing values using the other features.
- Remove records that are missing features.
- Use a machine learning technique that uses classification trees, e.g. Decision tree

282

innovate achieve lead

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, < X_1 … X_n >
  - Y is boolean
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik},\sigma_i)$
  - model P(Y) as Bernoulli (π)
- What does that imply about the form of P(Y|X)?

$$P(Y = 1|X =< X_1, ...X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

283

innovate achieve lead

Derive form for P(Y|X) for Gaussian $P(X_i \mid Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y = 1|X) = \frac{P(Y = 1)P(X|Y = 1)}{P(Y = 1)P(X|Y = 1) + P(Y = 0)P(X|Y = 0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp((\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

$$P(x \mid y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

$$P(Y = 1) = \pi$$

$$\sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2} \right)$$

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

284

Very convenient!

$$P(Y=1|X=<X_1,...X_n>)=\frac{1}{1+exp(w_0+\sum_i w_iX_i)}$$

implies

$$P(Y=0|X=<X_1,...X_n>)=\frac{exp(w_0+\sum_i w_iX_i)}{1+exp(w_0+\sum_i w_iX_i)}$$

implies

$$\frac{P(Y=0|X)}{P(Y=1|X)}=exp(w_0+\sum_i w_iX_i)$$

implies

$$\ln\frac{P(Y=0|X)}{P(Y=1|X)}=w_0+\sum_i w_iX_i$$

285

Very convenient!

$$P(Y=1|X=<X_1,...X_n>)=\frac{1}{1+exp(w_0+\sum_i w_iX_i)}$$

implies

$$P(Y=0|X=<X_1,...X_n>)=\frac{exp(w_0+\sum_i w_iX_i)}{1+exp(w_0+\sum_i w_iX_i)}$$

implies

$$\frac{P(Y=0|X)}{P(Y=1|X)}=exp(w_0+\sum_i w_iX_i)$$

linear classification rule!

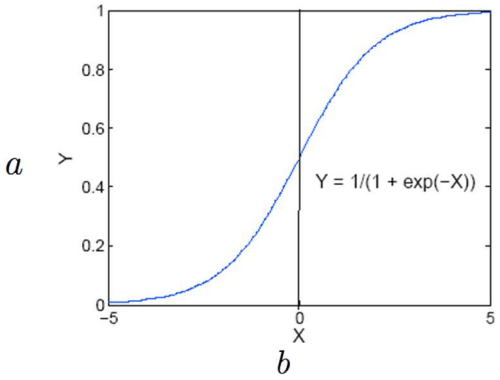
implies

$$\ln\frac{P(Y=0|X)}{P(Y=1|X)}=w_0+\sum_i w_iX_i$$

286

Logistic function

$$a = \frac{1}{1+\exp(-b)}$$



$$P(Y=1|X) = \frac{1}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

287

Logistic regression more generally

- Logistic regression when Y not boolean (but still discrete-valued).
- Now $y \in \{y_1 \ldots y_R\}$: learn R-1 sets of weights

for $k<R$ $$P(Y=y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^{n} w_{ki} X_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji} X_i)}$$

for $k=R$ $$P(Y=y_R|X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji} X_i)}$$

288

Training Logistic Regression: MCLE

- we have L training examples: $\{\langle X^1, Y^1\rangle, \ldots \langle X^L, Y^L\rangle\}$
- maximum likelihood estimate for parameters W

$$W_{MLE} = \arg\max_W P(< X^1, Y^1 > \ldots < X^L, Y^L > |W)$$
$$= \arg\max_W \prod_l P(< X^l, Y^l > |W)$$

- maximum conditional likelihood estimate

289

Training Logistic Regression: MCLE

- Choose parameters W=<w_0, ... w_n> to maximize conditional likelihood of training data

$$P(Y = 0|X, W) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

where

$$P(Y = 1|X, W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

- Training data D = $\{\langle X^1, Y^1\rangle, \ldots \langle X^L, Y^L\rangle\}$
- Data likelihood = $\prod_l P(X^l, Y^l|W)$
- Data conditional likelihood = $\prod_l P(Y^l|X^l, W)$

$$W_{MCLE} = \arg\max_W \prod_l P(Y^l|W, X^l)$$

290

Expressing Conditional Log Likelihood

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

$$P(Y=0|X,W) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$P(Y=1|X,W) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

$$\begin{aligned} l(W) &= \sum_l Y^l \ln P(Y^l=1|X^l,W) + (1-Y^l)\ln P(Y^l=0|X^l,W) \\ &= \sum_l Y^l \ln \frac{P(Y^l=1|X^l,W)}{P(Y^l=0|X^l,W)} + \ln P(Y^l=0|X^l,W) \\ &= \sum_l Y^l(w_0+\sum_i^n w_i X_i^l) - \ln(1+exp(w_0+\sum_i^n w_i X_i^l)) \end{aligned}$$

291

Maximizing Conditional Log Likelihood

$$P(Y=0|X,W) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$P(Y=1|X,W) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

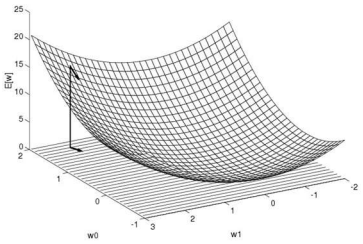
$$\begin{aligned} l(W) &\equiv \ln \prod_l P(Y^l|X^l,W) \\ &= \sum_l Y^l(w_0+\sum_i^n w_i X_i^l) - \ln(1+exp(w_0+\sum_i^n w_i X_i^l)) \end{aligned}$$

Good news: *l(W)* is concave function of *W*

Bad news: no closed-form solution to maximize *l(W)*

292

Gradient Descent



Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

293

Gradient Descent:

Batch gradient: use error $E_D(\mathbf{w})$ over entire training set D

Do until satisfied:

1. Compute the gradient $\nabla E_D(\mathbf{w}) = \left[\frac{\partial E_D(\mathbf{w})}{\partial w_0} \cdots \frac{\partial E_D(\mathbf{w})}{\partial w_n}\right]$
2. Update the vector of parameters: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E_D(\mathbf{w})$

Stochastic gradient: use error $E_d(\mathbf{w})$ single examples $d \in D$

Do until satisfied:

1. Choose (with replacement) a random training example $d \in D$
2. Compute the gradient just for d $\nabla E_d(\mathbf{w}) = \left[\frac{\partial E_d(\mathbf{w})}{\partial w_0} \cdots \frac{\partial E_d(\mathbf{w})}{\partial w_n}\right]$
3. Update the vector of parameters: $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E_d(\mathbf{w})$

Stochastic approximates Batch arbitrarily closely $\eta \rightarrow 0$ as Stochastic can be much faster when D is very large Intermediate approach: use error over subsets of D

294

Maximize Conditional Log Likelihood: Gradient Ascent

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W)$$
$$= \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l))$$

$$\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))$$

295

Maximize Conditional Log Likelihood: Gradient Ascent

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W)$$
$$= \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l))$$

$$\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))$$

Gradient ascent algorithm: iterate until change < ε

For all *i*, repeat

$$w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))$$

296

That's all for M(C)LE. How about MAP?

- One common approach is to define priors on W
  - Normal distribution, zero mean, identity covariance
- Helps avoid very large weights and overfitting
- MAP estimate

$$W \leftarrow \arg\max_W \ln P(W) \prod_l P(Y^l|X^l, W)$$

- let's assume Gaussian prior: W ~ N(0, σ)

297

MLE vs MAP

- Maximum conditional likelihood estimate

$$W \leftarrow \arg\max_W \ln \prod_l P(Y^l|X^l, W)$$

$$w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))$$

- Maximum a posteriori estimate with prior W~N(0,σI)

$$W \leftarrow \arg\max_W \ln[P(W) \prod_l P(Y^l|X^l, W)]$$

$$w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))$$

298

MAP estimates and Regularization

- Maximum a posteriori estimate with prior W~N(0,σI)

$$W \leftarrow \arg\max_W \ \ln[P(W) \prod_l P(Y^l|X^l,W)]$$

$$w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l,W))$$

λ is called a "regularization" term

- helps reduce overfitting
- keep weights nearer to zero (if P(W) is zero mean Gaussian prior), or whatever the prior suggests
- used very frequently in Logistic Regression

299

The Bottom Line

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, < X_1 … X_n >
  - Y is boolean
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik},\sigma_i)$
  - model P(Y) as Bernoulli (π)
- Then P(Y|X) is of this form, and we can directly estimate W

$$P(Y = 1|X =< X_1, \ldots X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

- Furthermore, same holds if the X_i are boolean
  - trying proving that to yourself

300

Logistic Regression Applications

- **Credit Card Fraud** : Predicting if a given credit card transaction is fraud or not
- **Health** : Predicting if a given mass of tissue is benign or malignant
- **Marketing** : Predicting if a given user will buy an insurance product or not
- **Banking** : Predicting if a customer will default on a loan.

301

Generative vs. Discriminative Classifiers

Training classifiers involves estimating f: X → Y, or P(Y|X)

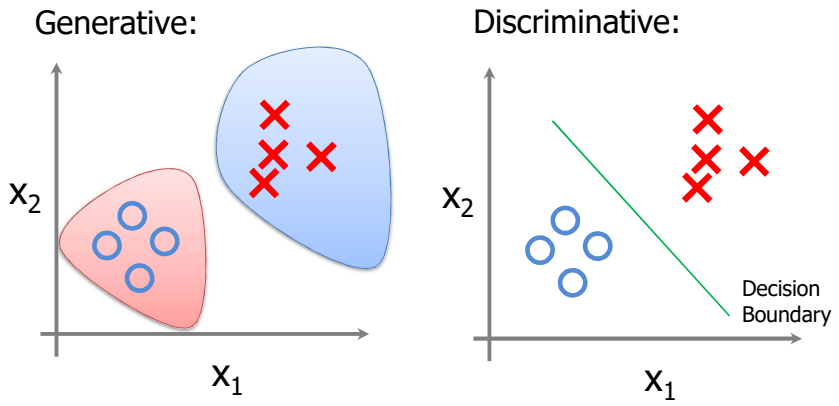
Generative classifiers (e.g., Naïve Bayes)
- Assume some functional form for P(X|Y), P(X)
- Estimate parameters of P(X|Y), P(X) directly from training data
- Use Bayes rule to calculate P(Y|X= x_i)

Discriminative classifiers (e.g., Logistic regression)

- Assume some functional form for P(Y|X)
- Estimate parameters of P(Y|X) directly from training data

302

Probabilistic Generative Model versus Probabilistic Discriminative Model



303

Probabilistic Generative Model versus Probabilistic Discriminative Model

Generative	Discriminative
Ex: Naïve Bayes	Ex: Logistic Regression

304

Use Naïve Bayes or Logistic Regression?

Consider

- Restrictiveness of modeling assumptions
- Rate of convergence (in amount of training data) toward asymptotic hypothesis

305

Naïve Bayes versus Logistic Regression

- Naïve Bayes are Generative Models which Logistic Regression are Discriminative Models
- Naïve Bayes easy to construct
- Naïve Bayes better on smaller datasets
- Naive Bayes also assumes that the features are conditionally independent. Real data sets are never perfectly independent
- When the training size reaches infinity, logistic regression performs better than the generative model Naive Bayes.
  - Optional reading by Ng and Jordan has proofs and experiments
- Logistic regression allows arbitrary features

306

Naïve Bayes vs Logistic Regression

innovate achieve lead

Consider Y boolean, X_i continuous, $X=<X_1 \ldots X_n>$

Number of parameters:

- NB: 4n +1
- LR: n+1

Estimation method:

- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

307

G. Naïve Bayes vs. Logistic Regression

innovate achieve lead

[Ng & Jordan, 2002]

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k)$ = $N(\mu_{ik},\sigma_i)$, ← not $N(\mu_{ik},\sigma_{ik})$

Consider three learning methods:

- GNB (assumption 1 only)
- GNB2 (assumption 1 and 2)
- LR

Which method works better if we have ***infinite*** training data, and...

- Both (1) and (2) are satisfied
- Neither (1) nor (2) is satisfied
- (1) is satisfied, but not (2)

308

G. Naïve Bayes vs. Logistic Regression

innovate achieve lead

[Ng & Jordan, 2002]

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k)$ = $N(\mu_{ik},\sigma_i)$, ← not $N(\mu_{ik},\sigma_{ik})$

Consider three learning methods:

- GNB (assumption 1 only) -- decision surface can be non-linear
- GNB2 (assumption 1 and 2) – decision surface linear
- LR -- decision surface linear, trained without assumption 1.

Which method works better if we have *infinite* training data, and...

- Both (1) and (2) are satisfied: LR = GNB2 = GNB
- (1) is satisfied, but not (2) : GNB > GNB2, GNB > LR, LR > GNB2
- Neither (1) nor (2) is satisfied: GNB>GNB2, LR > GNB2, LR><GNB

309

In our next session

innovate achieve lead

We will cover:

Linear basis function models
Bias-variance decomposition
Bayesian linear regression

310

innovate achieve lead

WEEK 6 - LINEAR REGRESSION

So far, we've been interested in learning P(Y|X) where Y has discrete values (called 'classification')

What if Y is continuous? (called 'regression')

- predict weight from gender, height, age, …
- predict Google stock price today from Google, Yahoo, MSFT prices yesterday
- predict each pixel intensity in robot's current camera image, from previous image and previous action

311

innovate achieve lead

Regression

Wish to learn f : X → Y , where Y is real, given $\{<x^1,y^1>\ldots<x^n,y^n>\}$

- Geometric
  - Least squares function fitting given $\{<x^1,y^1>\ldots<x^n,y^n>\}$

Bayesian

- Choose some parameterized form for P(Y|X; θ)

(θ is the vector of parameters)

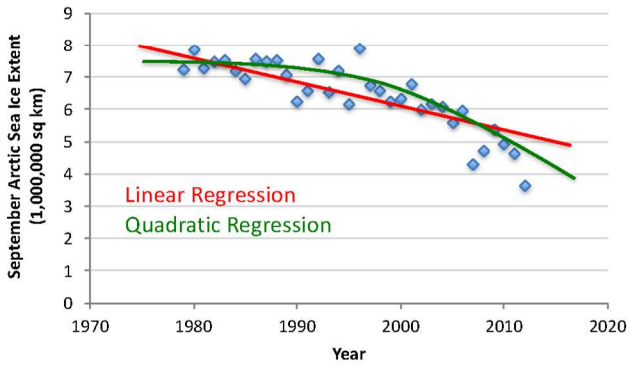
- Derive learning algorithm as MCLE or MAP estimate for θ

312

Geometric Approach

Given:

- Data $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\right\}$ where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$
- Corresponding labels $\boldsymbol{y} = \left\{y^{(1)}, \ldots, y^{(n)}\right\}$ where $y^{(i)} \in \mathbb{R}$



Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

2

313

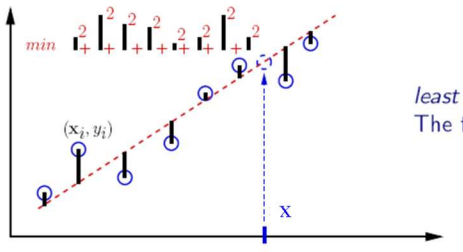
Linear Regression

- Hypothesis:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d = \sum_{j=0}^{d} \theta_j x_j$$

Assume $x_0 = 1$

- Fit model by minimizing sum of squared errors



least squares (LSQ)
The fitted line is used as a predictor

Figures are courtesy of Greg Shakhnarovich

3

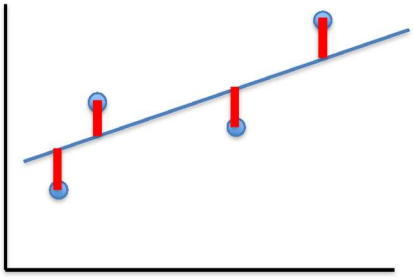
314

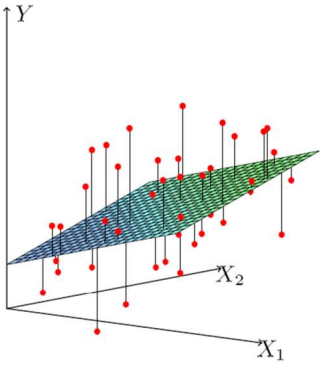
Least Squares Linear Regression

- Cost Function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

- Fit by solving $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$





4

315

Intuition Behind Cost Function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

For insight on J(), let's assume $x \in \mathbb{R}$ so $\boldsymbol{\theta} = [\theta_0, \theta_1]$

Based on example by Andrew Ng

5

316

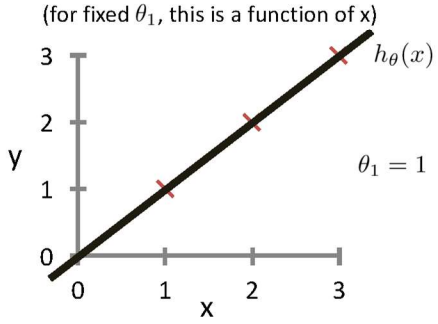
Intuition Behind Cost Function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

For insight on J(), let's assume $x \in \mathbb{R}$ so $\boldsymbol{\theta} = [\theta_0, \theta_1]$

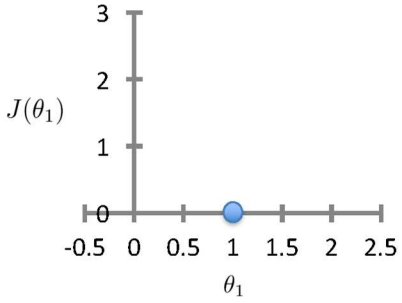
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)



Based on example by Andrew Ng

317

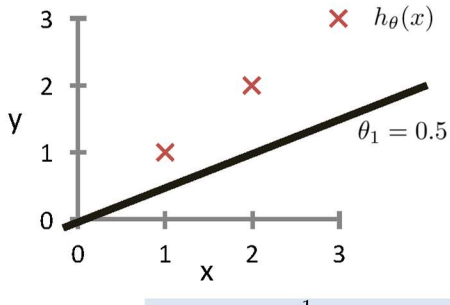
Intuition Behind Cost Function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

For insight on J(), let's assume $x \in \mathbb{R}$ so $\boldsymbol{\theta} = [\theta_0, \theta_1]$

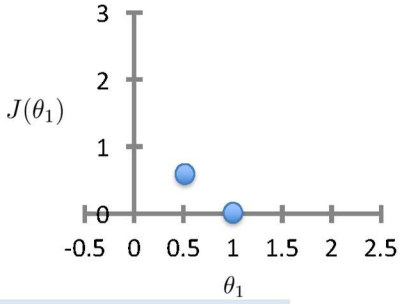
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)



Based on example by Andrew Ng

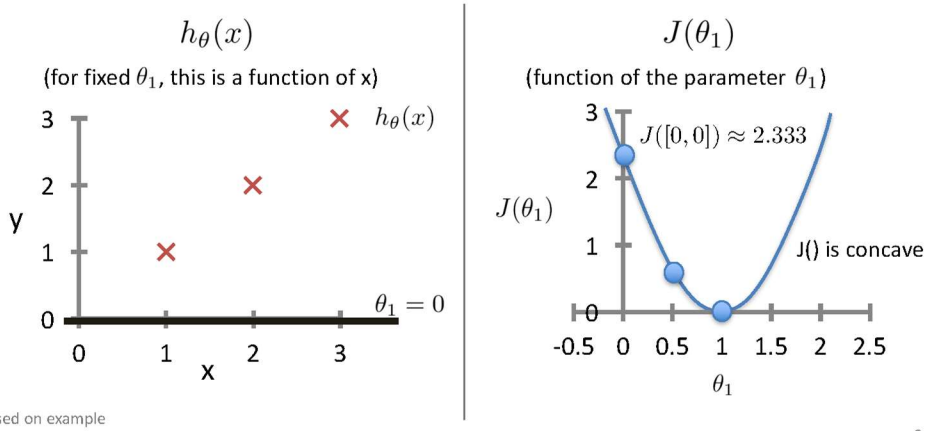
$$J([0, 0.5]) = \frac{1}{2\times 3}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right] \approx 0.58$$

318

Intuition Behind Cost Function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

For insight on J(), let's assume $x \in \mathbb{R}$ so $\boldsymbol{\theta} = [\theta_0, \theta_1]$

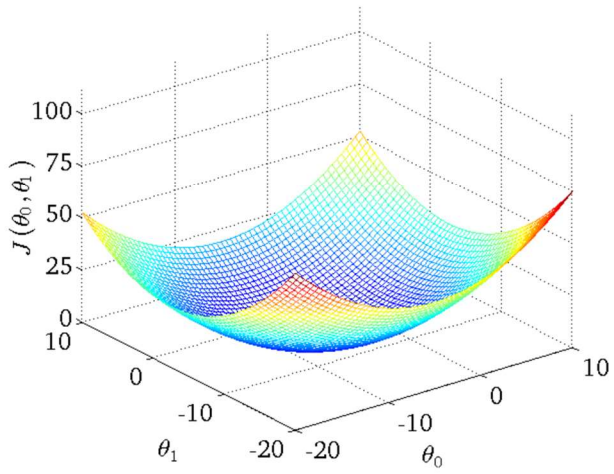


Based on example by Andrew Ng

8

319

Intuition Behind Cost Function



Slide by Andrew Ng

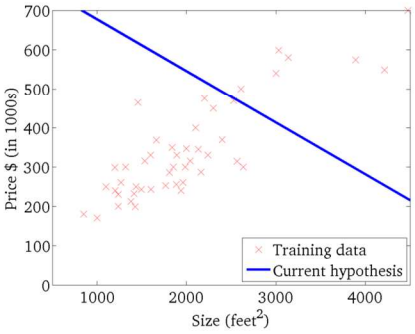
9

320

Intuition Behind Cost Function

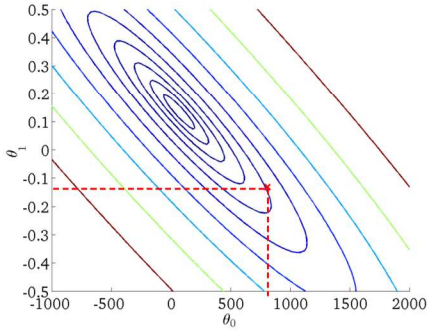
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



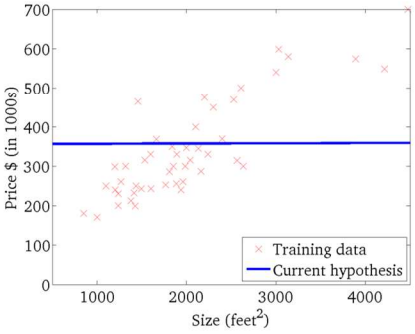
Slide by Andrew Ng 10

321

Intuition Behind Cost Function

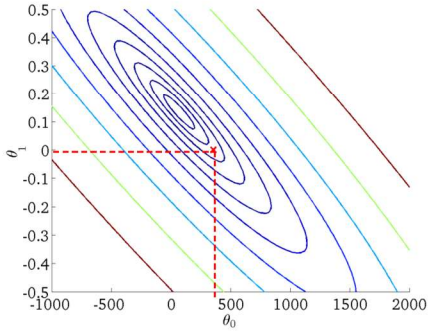
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



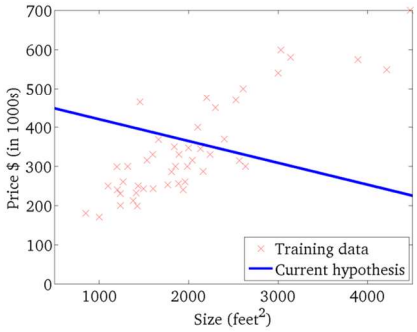
Slide by Andrew Ng 11

322

Intuition Behind Cost Function

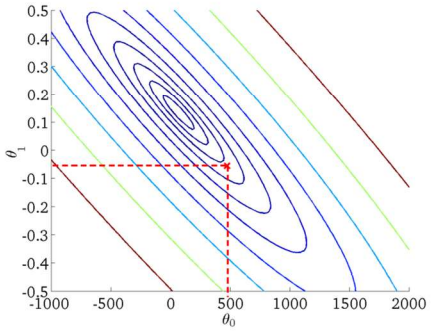
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



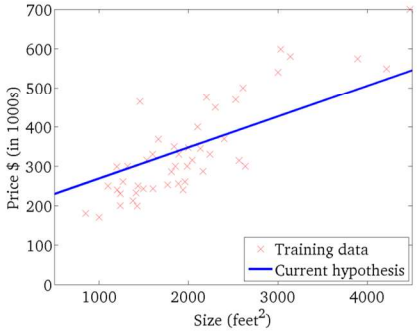
Slide by Andrew Ng 12

323

Intuition Behind Cost Function

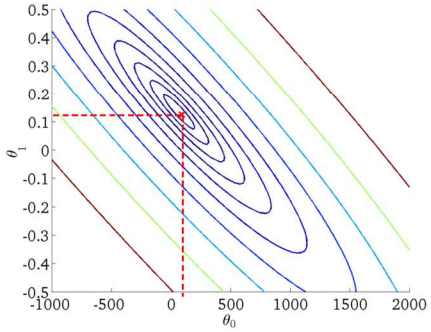
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



Slide by Andrew Ng 13

324

Basic Search Procedure

- Choose initial value for θ
- Until we reach a minimum:
  - Choose a new value for θ to reduce $J(\theta)$

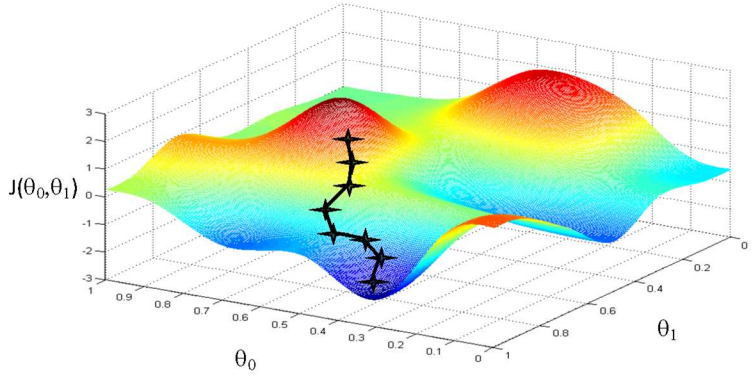


Figure by Andrew Ng 14

325

Basic Search Procedure

- Choose initial value for θ
- Until we reach a minimum:
  - Choose a new value for θ to reduce $J(\theta)$

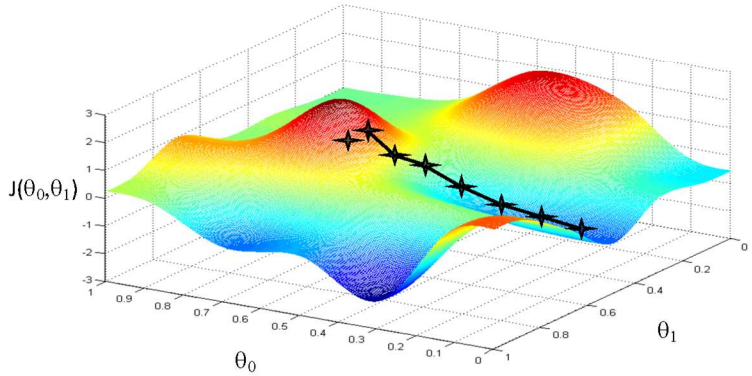
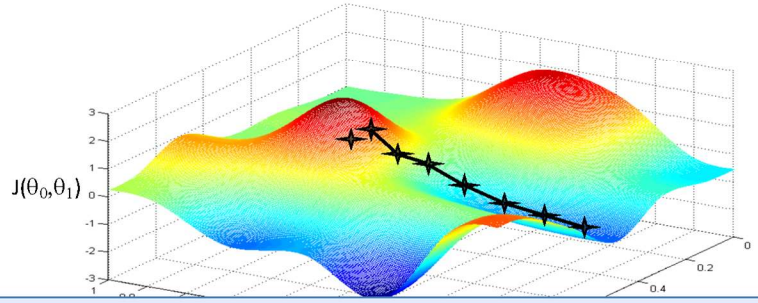


Figure by Andrew Ng 15

326

Basic Search Procedure

- Choose initial value for $\boldsymbol{\theta}$
- Until we reach a minimum:
  - Choose a new value for $\boldsymbol{\theta}$ to reduce $J(\boldsymbol{\theta})$



Since the least squares objective function is convex (concave), we don't need to worry about local minima

Figure by Andrew Ng 16

327

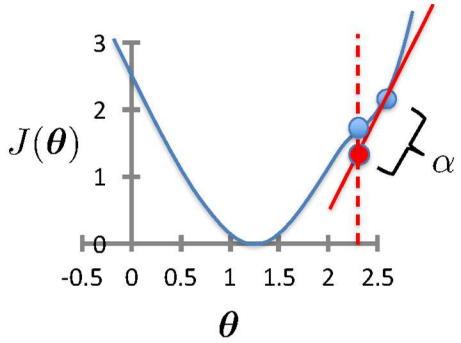
Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update for j = 0 ... d

learning rate (small) e.g., α = 0.05



17

328

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update for j = 0 ... d

For Linear Regression: $\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2$

18

329

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update for j = 0 ... d

For Linear Regression: $\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2$$

19

330

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update for j = 0 ... d

For Linear Regression:

$$\begin{aligned}\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)} \right)^2 \\ &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \times \frac{\partial}{\partial \theta_j} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)\end{aligned}$$

20

331

Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update for j = 0 ... d

For Linear Regression:

$$\begin{aligned}\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)} \right)^2 \\ &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \times \frac{\partial}{\partial \theta_j} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) x_j^{(i)}\end{aligned}$$

21

332

Gradient Descent for Linear Regression

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right) x_j^{(i)}$$

simultaneous update for j = 0 ... d

- To achieve simultaneous update
  - At the start of each GD iteration, compute $h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right)$
  - Use this stored value in the update step loop
- Assume convergence when $\|\boldsymbol{\theta}_{new} - \boldsymbol{\theta}_{old}\|_2 < \epsilon$

L_2 norm: $\|\boldsymbol{v}\|_2 = \sqrt{\sum_i v_i^2} = \sqrt{v_1^2 + v_2^2 + \ldots + v_{|v|}^2}$

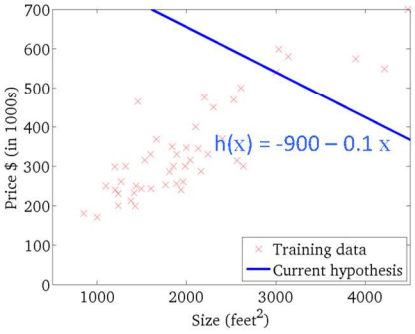
22

333

Gradient Descent

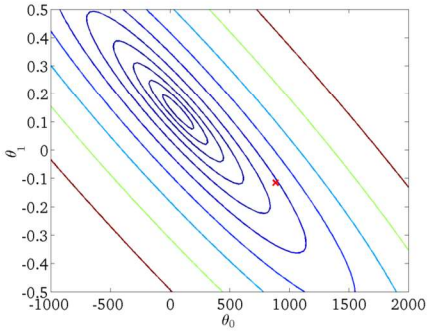
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



Slide by Andrew Ng

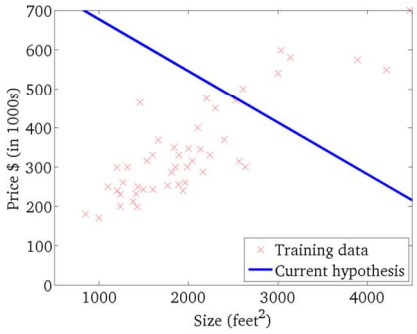
23

334

Gradient Descent

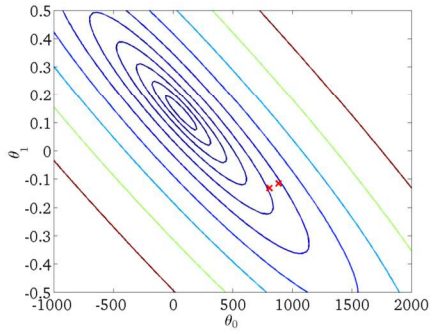
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



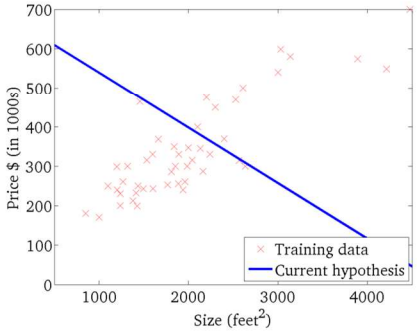
Slide by Andrew Ng 24

335

Gradient Descent

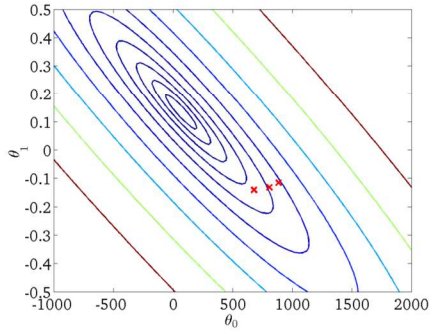
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



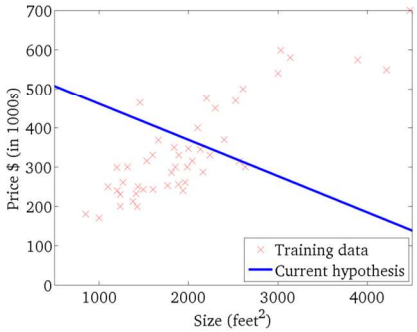
Slide by Andrew Ng 25

336

Gradient Descent

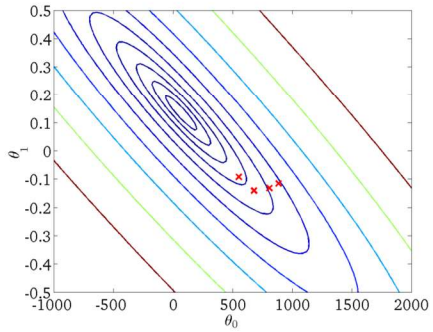
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



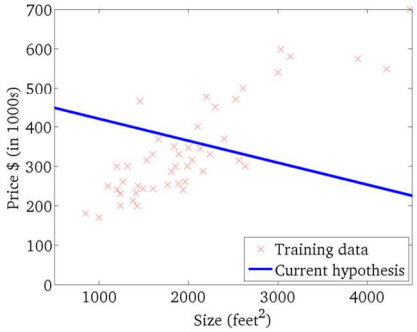
Slide by Andrew Ng 26

337

Gradient Descent

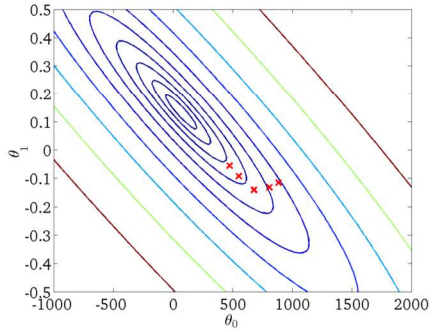
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



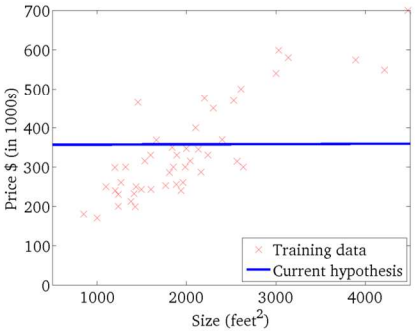
Slide by Andrew Ng 27

338

Gradient Descent

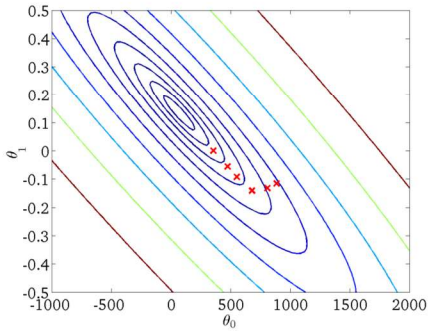
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



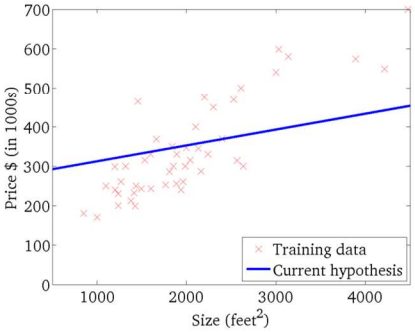
Slide by Andrew Ng 28

339

Gradient Descent

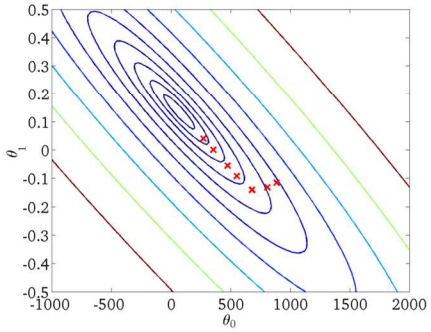
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



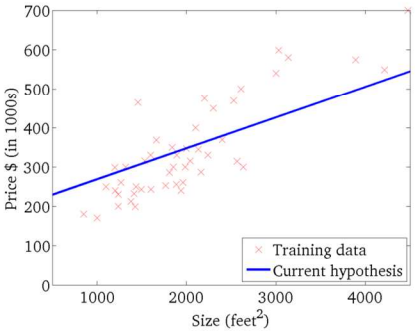
Slide by Andrew Ng 29

340

Gradient Descent

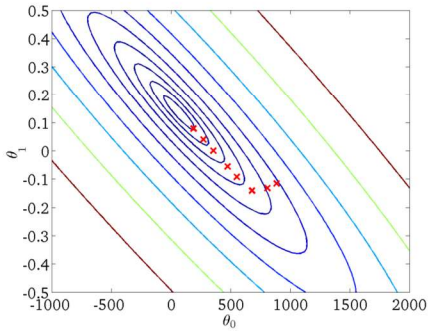
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



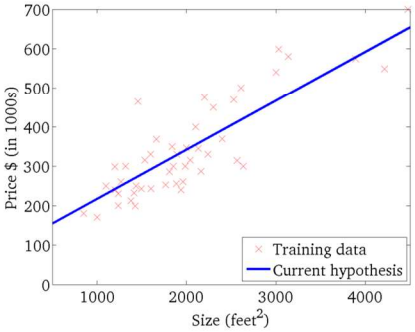
Slide by Andrew Ng 30

341

Gradient Descent

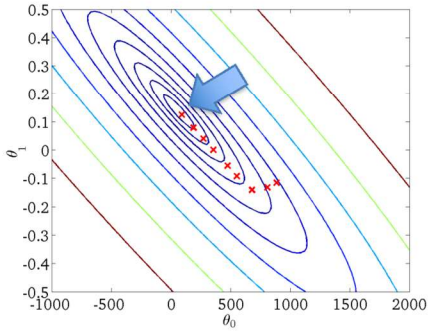
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

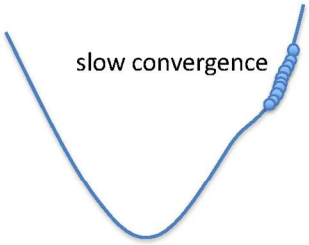


Slide by Andrew Ng 31

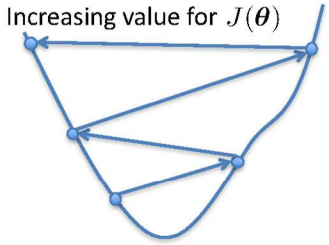
342

Choosing α

α too small



α too large



- May overshoot the minimum
- May fail to converge
- May even diverge

To see if gradient descent is working, print out $J(\theta)$ each iteration
- The value should decrease at each iteration
- If it doesn't, adjust α

32

343

innovate achieve lead

Learning Model Parameters – Closed Form Solution (using vectorization)

344

BITS Pilani, Pilani Campus

344

Vectorization

- Benefits of vectorization
  - More compact equations
  - Faster code (using optimized matrix libraries)
- Consider our model:

$$h(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j x_j$$

- Let

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \qquad \boldsymbol{x}^\intercal = \begin{bmatrix} 1 & x_1 & \dots & x_d \end{bmatrix}$$

- Can write the model in vectorized form as $h(\boldsymbol{x}) = \boldsymbol{\theta}^\intercal \boldsymbol{x}$

43

345

Vectorization

- Consider our model for n instances:

$$h\left(\boldsymbol{x}^{(i)}\right) = \sum_{j=0}^{d} \theta_j x_j^{(i)}$$

- Let

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(i)} & \dots & x_d^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \dots & x_d^{(n)} \end{bmatrix}$$

$\mathbb{R}^{(d+1)\times 1}$ $\mathbb{R}^{n\times(d+1)}$

- Can write the model in vectorized form as $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{X}\boldsymbol{\theta}$

44

346

Vectorization

- For the linear regression cost function:

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}^{(i)} - y^{(i)}\right)^2$$

$$= \frac{1}{2n}\underbrace{(\boldsymbol{X\theta} - \boldsymbol{y})^\intercal}_{\mathbb{R}^{1\times n}}\underbrace{(\boldsymbol{X\theta} - \boldsymbol{y})}_{\mathbb{R}^{n\times 1}}$$

$\mathbb{R}^{n\times(d+1)}$, $\mathbb{R}^{(d+1)\times 1}$

Let:

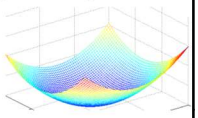
$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

45

347

Closed Form Solution

- Instead of using GD, solve for optimal $\boldsymbol{\theta}$ analytically
  - Notice that the solution is when $\frac{\partial}{\partial \boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0$



- Derivation:

$$\mathcal{J}(\boldsymbol{\theta}) = \frac{1}{2n}(\boldsymbol{X\theta} - \boldsymbol{y})^\intercal(\boldsymbol{X\theta} - \boldsymbol{y})$$

$$\propto \boldsymbol{\theta}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X\theta} - \boxed{\boldsymbol{y}^\intercal \boldsymbol{X\theta}} - \boxed{\boldsymbol{\theta}^\intercal \boldsymbol{X}^\intercal \boldsymbol{y}} + \boldsymbol{y}^\intercal \boldsymbol{y}$$

1 x 1

$$\propto \boldsymbol{\theta}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X\theta} - 2\boldsymbol{\theta}^\intercal \boldsymbol{X}^\intercal \boldsymbol{y} + \boldsymbol{y}^\intercal \boldsymbol{y}$$

Take derivative and set equal to 0, then solve for $\boldsymbol{\theta}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X\theta} - 2\boldsymbol{\theta}^\intercal \boldsymbol{X}^\intercal \boldsymbol{y} + \cancel{\boldsymbol{y}^\intercal \boldsymbol{y}}) = 0$$

$$(\boldsymbol{X}^\intercal \boldsymbol{X})\boldsymbol{\theta} - \boldsymbol{X}^\intercal \boldsymbol{y} = 0$$

$$(\boldsymbol{X}^\intercal \boldsymbol{X})\boldsymbol{\theta} = \boldsymbol{X}^\intercal \boldsymbol{y}$$

Closed Form Solution: $\boldsymbol{\theta} = (\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}\boldsymbol{X}^\intercal \boldsymbol{y}$

46

348

Closed Form Solution

- Can obtain $\boldsymbol{\theta}$ by simply plugging $\boldsymbol{X}$ and $\boldsymbol{y}$ into

$$\boldsymbol{\theta} = (\boldsymbol{X}^\intercal \boldsymbol{X})^{-1} \boldsymbol{X}^\intercal \boldsymbol{y}$$

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(i)} & \dots & x_d^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \dots & x_d^{(n)} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

- If $\boldsymbol{X}^\intercal \boldsymbol{X}$ is not invertible (i.e., singular), may need to:
  - Use pseudo-inverse instead of the inverse
    - In python, `numpy.linalg.pinv(a)`
  - Remove redundant (not linearly independent) features
  - Remove extra features to ensure that d ≤ n

47

349

Gradient Descent vs Closed Form

Gradient Descent	Closed Form Solution
• Requires multiple iterations • Need to choose α • Works well when n is large • Can support incremental learning	• Non-iterative • No need for α • Slow if n is large – Computing $(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}$ is roughly $O(n^3)$

48

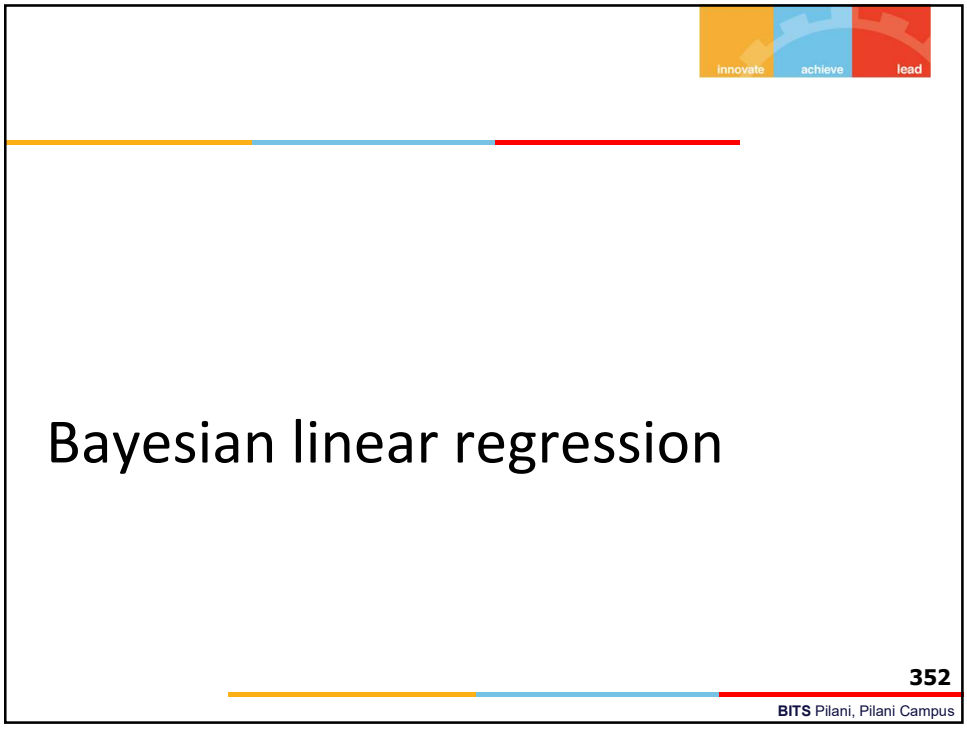
350

Extending Linear Regression to More Complex Models

- The inputs **X** for linear regression can be:
  - Original quantitative inputs
  - Transformation of quantitative inputs
    - e.g. log, exp, square root, square, etc.
  - Polynomial transformation
    - example: $y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$
  - Basis expansions
  - Dummy coding of categorical inputs
  - Interactions between variables
    - example: $x_3 = x_1 \cdot x_2$

This allows use of linear regression techniques to fit non-linear datasets.

351



352

Bayesian analysis

- Bayesian analysis will show that
  - under certain assumptions any learning algorithm that minimizes the squared error between the output hypothesis predictions and the training data will output a maximum likelihood hypothesis

353

Maximum likelihood and least-squared error hypotheses

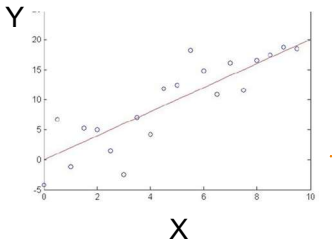
- A set of m training examples is provided, where the target value of each example is corrupted by random noise drawn according to a Normal probability distribution.
- Each training example is a pair of the form

(x_i, d_i) where $d_i = f(x_i) + e_i$. Here $f(x_i)$ is the noise-free value of the target function and e_i is a random variable representing the noise.

  - values of the e_i are drawn independently and that they are distributed according to a Normal distribution with zero mean

354

Choose parameterized form for P(Y|X; θ)



Assume Y is some deterministic f(X), plus random noise

$$y = f(x) + \epsilon \quad \text{where} \quad \epsilon \sim N(0, \sigma)$$

Therefore Y is a random variable that follows the distribution

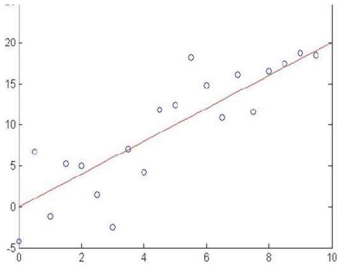
$$p(y|x) = N(f(x), \sigma)$$

and the expected value of y for any given x is f(x)

355

Consider Linear Regression

$$p(y|x) = N(f(x), \sigma)$$



E.g., assume f(x) is linear function of x

$$p(y|x) = N(w_0 + w_1 x, \sigma)$$

$$E[y|x] = w_0 + w_1 x$$

Notation: to make our parameters explicit, let's write

$$W =< w_0, w_1 >$$

$$p(y|x; W) = N(w_0 + w_1 x, \sigma)$$

356

Training Linear Regression : Maximum Conditional Likelihood Estimate (MCLE)

$$p(y|x;W) = N(w_0 + w_1 x, \sigma)$$

How can we learn W from the training data?

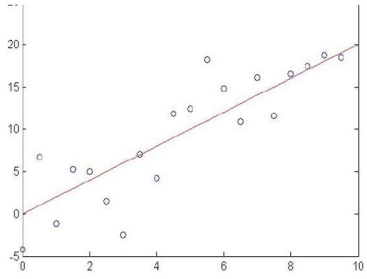
Learn Maximum Conditional Likelihood Estimate!

$$W_{MCLE} = \arg\max_W \prod_l p(y^l|x^l, W)$$

$$W_{MCLE} = \arg\max_W \sum_l \ln p(y^l|x^l, W)$$

where

$$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}(\frac{y-f(x;W)}{\sigma})^2}$$



357

Training Linear Regression: MCLE

Learn Maximum Conditional Likelihood Estimate

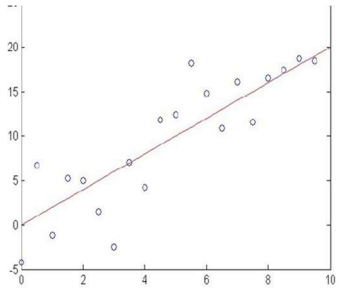
$$W_{MCLE} = \arg\max_W \sum_l \ln p(y^l|x^l, W)$$

where

$$p(y|x;W) = \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}(\frac{y-f(x;W)}{\sigma})^2}$$

so:

$$W_{MCLE} = \arg\min_W \sum_l (y - f(x;W))^2$$



358

$$h_{ML} = \underset{h \in H}{\text{argmax}} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

The first term in this expression is a constant independent of h, and can therefore be discarded, yielding

$$h_{ML} = \underset{h \in H}{\text{argmax}} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

Maximizing this negative quantity is equivalent to minimizing the corresponding positive quantity.

$$h_{ML} = \underset{h \in H}{\text{argmin}} \sum_{i=1}^{m} \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

Finally, we can again discard constants that are independent of h.

$$h_{ML} = \underset{h \in H}{\text{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2 \quad (6.6)$$

359

359

Training Linear Regression

Maximum Conditional Likelihood Estimate is equivalent to minimizing the squared error loss

$$W_{MCLE} = \arg\min_{W} \sum_{l} (y - f(x; W))^2$$

Can we derive gradient descent rule for training?

$$\frac{\partial \sum_l (y - f(x;W))^2}{\partial w_i} = \sum_l 2(y - f(x;W)) \frac{\partial (y - f(x;W))}{\partial w_i}$$

$$= \sum_l -2(y - f(x;W)) \frac{\partial f(x;W)}{\partial w_i}$$

360

Regression – What you should know

Under general assumption

$$p(y|x;W) = N(f(x;W), \sigma)$$

- We can use gradient descent as a general learning algorithm
  - as long as our objective fn is differentiable wrt W
  - though we might learn local optima
- Almost nothing we said here required that f(x) be linear in x

361

Linear Basis Function Models

362

Linear Basis Function

- Simplest linear model for regression is one that involves a linear combination of the input variables

$$y(x,w)=w_0 +w_1x_1 +...+w_Dx_D$$

- Extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form

$$y(x,w)=w_0 + \sum_{j=1}^{M-1} w_j\varphi_j(x)$$

  - where $\phi_j(x)$ are known as basis functions.
  - By denoting the maximum value of the index j by M −1, the total number of parameters in this model will be M.

363

Linear Basis Function

- Convenient to define an additional dummy 'basis function' $\phi_0(x)=1$. So,

$$y(x,w) = \sum_{j=1}^{M-1} w_j\varphi_j(x) = \mathbf{w}^T \boldsymbol{\phi}_j(\mathbf{x})$$

where $\mathbf{w} =(w_0,...,w_{M-1})^T$ and $\boldsymbol{\phi} =(\phi_0,\phi_1,.....\phi_n)$

- If the original variables comprise the vector x, then the features can be expressed in terms of the basis functions {ϕj(x)}

364

Linear Basis Function Models

- Generally,

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j \underbrace{\phi_j(\boldsymbol{x})}_{\text{basis function}}$$

- Typically, $\phi_0(\boldsymbol{x}) = 1$ so that θ_0 acts as a bias
- In the simplest case, we use linear basis functions :

$$\phi_j(\boldsymbol{x}) = x_j$$

Based on slide by Christopher Bishop (PRML)

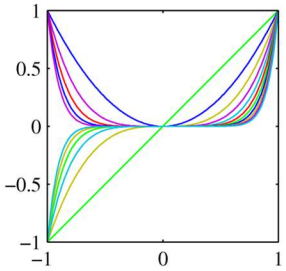
365

Linear Basis Function Models

- Polynomial basis functions:

$$\phi_j(x) = x^j$$

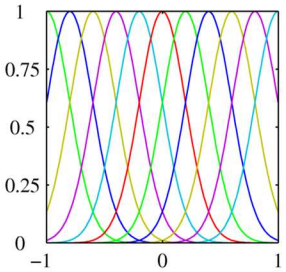
  - These are global; a small change in x affects all basis functions



- Gaussian basis functions:

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

  - These are local; a small change in x only affect nearby basis functions. μ_j and s control location and scale (width).



Based on slide by Christopher Bishop (PRML)

366

Linear Basis Function Models

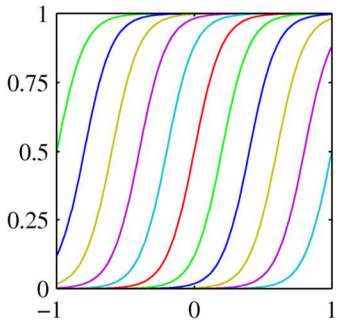
- Sigmoidal basis functions:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

  - These are also local; a small change in x only affects nearby basis functions. μ_j and s control location and scale (slope).



Based on slide by Christopher Bishop (PRML)

367

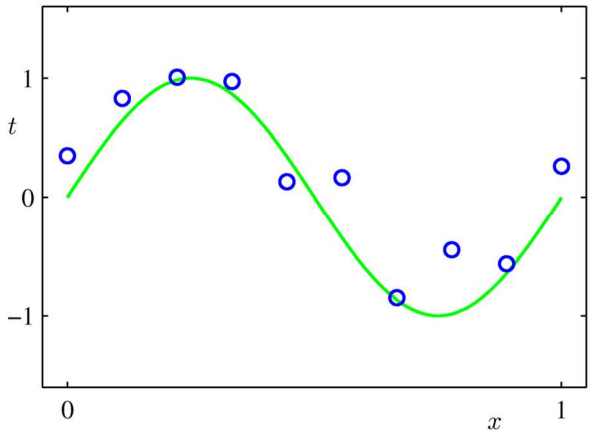
Linear Basis Function Models

- Basic Linear Model: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j x_j$
- Generalized Linear Model: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j \phi_j(\boldsymbol{x})$
- Once we have replaced the data by the outputs of the basis functions, fitting the generalized model is exactly the same problem as fitting the basic model
  - Unless we use the kernel trick – more on that when we cover support vector machines
  - Therefore, there is no point in cluttering the math with basis functions

Based on slide by Geoff Hinton 38

368

Example of Fitting a Polynomial Curve with a Linear Model



$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p = \sum_{j=0}^{p} \theta_j x^j$$

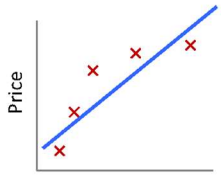
369

innovate achieve lead

Bias-Variance Decomposition

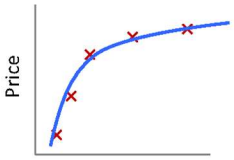
370

Quality of Fit



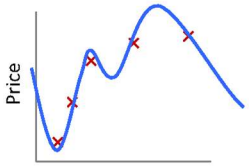
$\theta_0 + \theta_1 x$

Underfitting
(high bias)



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

Correct fit



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

Overfitting
(high variance)

Overfitting:

- The learned hypothesis may fit the training set very well ($J(\boldsymbol{\theta}) \approx 0$)
- ...but fails to generalize to new examples

Based on example by Andrew Ng

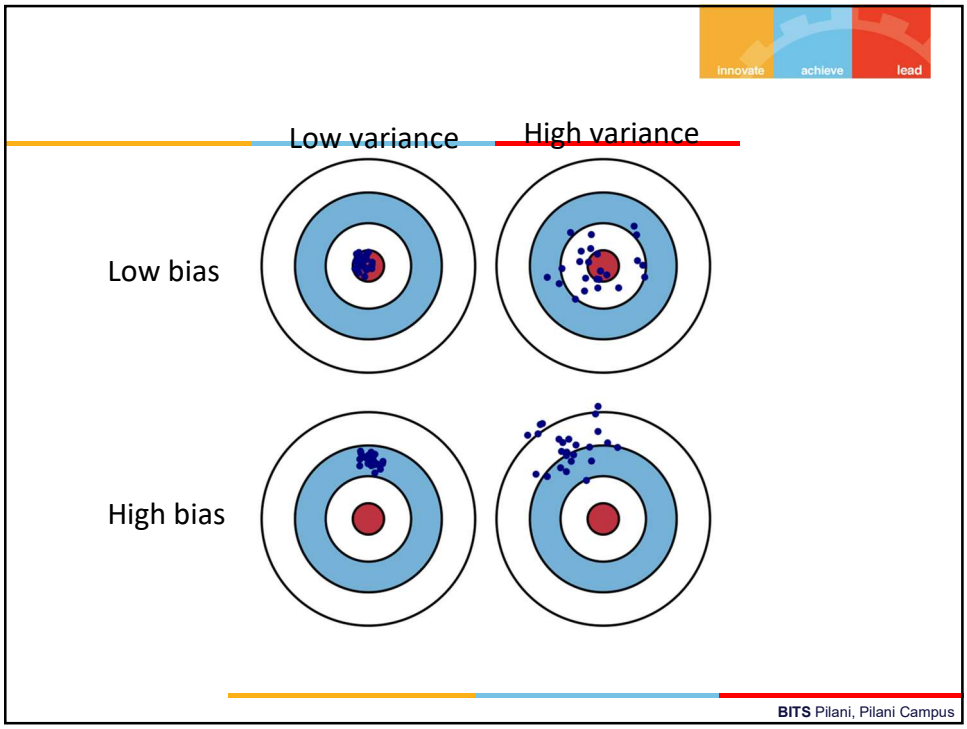
51

371

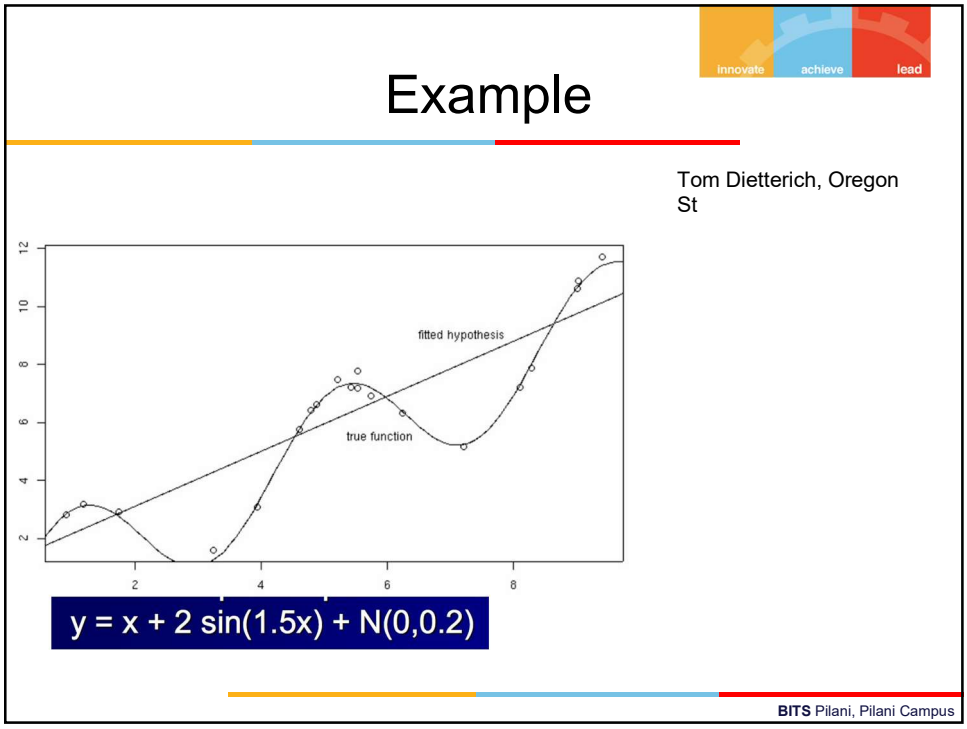
Bias-Variance Tradeoff

- **Bias**: difference between
  what you expect to learn and **truth**
  - Measures how well you expect to represent true solution
  - Decreases with more complex model
- **Variance**: difference between
  what you expect to learn and
  what you learn from a particular dataset
  - Measures how sensitive learner is to specific dataset
  - Increases with more complex model

372


innovate
achieve
lead
Low variance
High variance
Low bias
High bias
BITS Pilani, Pilani Campus

373


innovate
achieve
lead
Example
Tom Dietterich, Oregon St
fitted hypothesis
true function
y = x + 2 sin(1.5x) + N(0,0.2)
BITS Pilani, Pilani Campus

374

Bias – Variance decomposition of error

$$E_{D,\varepsilon}\{(f(x)+\varepsilon-h_D)^2\}$$

dataset and noise

true function

noise

learned from D

Fix test case *x*, then do this experiment:

1. Draw size *n* sample D=$(x_1,y_1),\ldots.(x_n,y_n)$
2. Train linear regressor h_D *using* D
3. Draw one test example *(x, f(x)+ε)*
4. Measure squared error of h_D on that example *x*
5. What's the expected error?

375

Bias – Variance decomposition of error

Notation - to simplify this

$$f \equiv f(x) + \varepsilon \qquad \hat{y} = \hat{y}_D \equiv h_D(x)$$

$$E_{D,\varepsilon}\{(f(x)+\varepsilon-h_D(x))^2\}$$

dataset and noise

true function

noise

learned from D

$$h \equiv E_D\{h_D(x)\}$$

long-term expectation of learner's prediction on this *x* averaged over many data sets *D*

376

Bias – Variance decomposition of error

$$E_{D,\varepsilon}\{(f - \hat{y})^2\}$$
$$= E\{([f - h] + [h - \hat{y}])^2\}$$
$$= E\{[f - h]^2 + [h - \hat{y}]^2 + 2[f - h][h - \hat{y}]\}$$
$$= E\{[f - h]^2 + [h - \hat{y}]^2 + 2[fh - f\hat{y} - h^2 + h\hat{y}]\}$$
$$= E[(f - h)^2] + E[(h - \hat{y})^2] + 2(E[fh] - E[f\hat{y}] - E[h^2] + E[h\hat{y}])$$

$$h \equiv E_D\{h(x)\}$$
$$h = E_D\{\hat{y}\}$$
$$\hat{y} = h_D \equiv h$$
$$f \equiv f(x) + \varepsilon$$

377

Bias – Variance decomposition of error

$$E_{D,\varepsilon}\{(f - \hat{y})^2\}$$
$$= E\{([f - h] + [h - \hat{y}])^2\}$$
$$= E\{[f - h]^2 + [h - \hat{y}]^2 + 2[f - h][h - \hat{y}]\}$$
$$= E[(f - h)^2] + E[(h - \hat{y})^2]$$

$$h \equiv E_D\{h(x)\}$$
$$\hat{y} = y_D(x) \equiv h_D(x)$$
$$f \equiv f(x) + \varepsilon$$

BIAS²: Squared difference between best possible prediction for x, f(x), and our "long-term" expectation for what the learner will do if we averaged over many datasets D, $E_D[h_D(x)]$

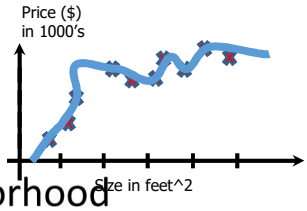
VARIANCE: Squared difference btwn our long-term expectation for the learners performance, $E_D[h_D(x)]$, and what we expect in a representative run on a dataset D (hat y)

13

378

Addressing overfitting

- x_1 = size of house
- x_2 = no. of bedrooms
- x_3 = no. of floors
- x_4 = age of house
- x_5 = average income in neighborhood
- x_6 = kitchen size
- ⋮
- x_{100}



Slide credit: Andrew Ng

379

Addressing overfitting

- **1. Reduce number of features.**
  - Manually select which features to keep.
  - Model selection algorithm (later in course).
- **2. Regularization.**
  - Keep all the features, but reduce magnitude/values θ_j.
  - Works well when we have a lot of features, each of contributes a bit to predicting y.

Slide credit: Andrew Ng

380

Regularization

- A method for automatically controlling the complexity of the learned hypothesis
- **Idea**: penalize for large values of θ_j
  - Can incorporate into the cost function
  - Works well when we have a lot of features, each that contributes a bit to predicting the label
- Can also address overfitting by eliminating features (either manually or via model selection)

52

381

Regularization

- Linear regression objective function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

model fit to data regularization

  - λ is the regularization parameter ($\lambda \geq 0$)
  - No regularization on θ_0!

53

382

Understanding Regularization

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

- Note that $\sum_{j=1}^{d}\theta_j^2 = \|\boldsymbol{\theta}_{1:d}\|_2^2$
  - This is the magnitude of the feature coefficient vector!
- We can also think of this as:

$$\sum_{j=1}^{d}(\theta_j - 0)^2 = \|\boldsymbol{\theta}_{1:d} - \vec{\mathbf{0}}\|_2^2$$

  - L_2 regularization pulls coefficients toward 0

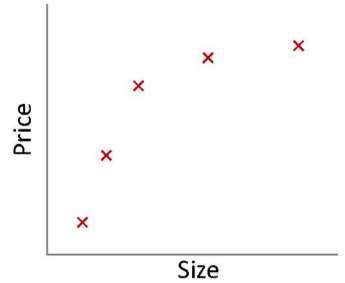
54

383

Understanding Regularization

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

- What happens if we set λ to be huge (e.g., 10^{10})?



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

Based on example by Andrew Ng

55

384

Understanding Regularization

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

- What happens if we set λ to be huge (e.g., 10^{10})?

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

0 0 0 0

Based on example by Andrew Ng

56

385

Regularized Linear Regression

- Cost Function

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

- Fit by solving $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
- Gradient update:

$\frac{\partial}{\partial\theta_0}J(\theta)$

$$\theta_0 \leftarrow \theta_0 - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)$$

$\frac{\partial}{\partial\theta_j}J(\theta)$

$$\theta_j \leftarrow \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} - \lambda\theta_j$$

regularization

57

386

Regularized Linear Regression

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

$$\theta_0 \leftarrow \theta_0 - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)$$

$$\theta_j \leftarrow \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} - \lambda\theta_j$$

- We can rewrite the gradient step as:

$$\theta_j \leftarrow \theta_j\left(1 - \alpha\lambda\right) - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

58

387

innovate achieve lead

WEEK 7 - Decision trees

- Decision Trees is one of the most widely used and practical methods of classification
- Method for approximating discrete-valued functions
- Learned functions are represented as decision trees (or if-then-else rules)
- Expressive hypotheses space

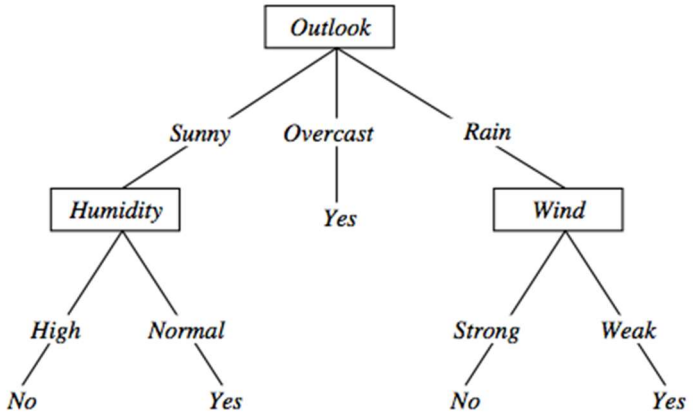
388

Decision Tree

- Advantages:
– Inexpensive to construct
– Extremely fast at classifying unknown records
– Easy to interpret for small-sized trees
– Can easily handle redundant or irrelevant attributes (unless the attributes are interacting)
- Disadvantages:
– Space of possible decision trees is exponentially large.
Greedy approaches are often unable to find the best tree.
– Does not take into account interactions between attributes
– Each decision boundary involves only a single attribute

389

Decision tree representation (PlayTennis)



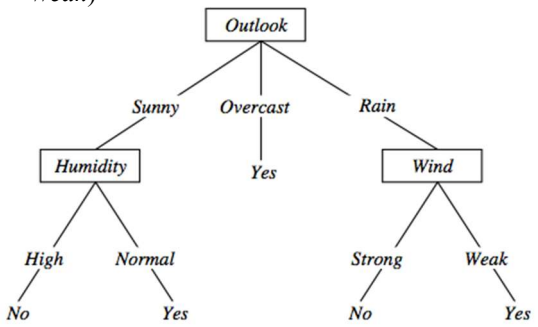
⟨*Outlook=Sunny, Temp=Hot, Humidity=High, Wind=Strong*⟩ No

390

Decision trees expressivity

- Decision trees represent a disjunction of conjunctions on constraints on the value of attributes:

$(Outlook = Sunny \wedge Humidity = Normal) \vee$
$(Outlook = Overcast) \vee$
$(Outlook = Rain \wedge Wind = Weak)$



391

Measure of Information

- The amount of information (surprise element) conveyed by a message is inversely proportional to its probability of occurrence. That is

$$I_k \alpha \frac{1}{p_k}$$

- The mathematical operator satisfies above properties is the logarithmic operator.

$$I_k = \log_r \frac{1}{p_k} \; units$$

392

Entropy

- Entropy of discrete random variable X={x_1, x_2…x_n}

$$\mathrm{H}(X) = \mathrm{E}[\mathrm{I}(X)] = \mathrm{E}[-\log(\mathrm{P}(X))].$$

; since: $\log_2(1/P(\text{event})) = -\log_2 P(\text{event})$

- As uncertainty increases, entropy increases
- Entropy across all values

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

393

Entropy in general

- Entropy measures the amount of information in a random variable

$H(X) = -p_+ \log_2 p_+ - p_- \log_2 p_-$ $\quad X = \{+, -\}$

for binary classification [two-valued random variable]

$$H(X) = -\sum_{i=1}^{c} p_i \log_2 p_i = \sum_{i=1}^{c} p_i \log_2 1/p_i \quad X = \{i, \ldots, c\}$$

for classification in *c* classes

394

Entropy in binary classification

- Entropy measures the *impurity* of a collection of examples. It depends from the distribution of the random variable *p.*
  - S is a collection of training examples
  - p_+ the proportion of positive examples in S
  - p_- the proportion of negative examples in S

$Entropy\ (S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_- \quad [0 \log_2 0 = 0]$

$Entropy\ ([14+, 0-]) = -14/14 \log_2 (14/14) - 0 \log_2 (0) = 0$

$Entropy\ ([9+, 5-]) = -9/14 \log_2 (9/14) - 5/14 \log_2 (5/14) = 0.94$

$Entropy\ ([7+, 7-]) = -7/14 \log_2 (7/14) - 7/14 \log_2 (7/14) =$

$= 1/2 + 1/2 = 1 \quad [\log_2 1/2 = -1]$

Note: the log of a number < 1 is negative, $0 \le p \le 1$, $0 \le entropy \le 1$

- https://www.easycalculation.com/log-base2-calculator.php

395

Information gain as entropy reduction

- *Information gain* is the *expected* reduction in entropy caused by partitioning the examples on an attribute.
- The higher the information gain the more effective the attribute in classifying training data.
- Expected reduction in entropy knowing *A*

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|Sv|}{|S|} Entropy(S_v)$$

$Values(A)$ possible values for A

S_v subset of S for which A has value v

396

Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

397

Example: Information gain

- Let
  - $Values(Wind) = \{Weak, Strong\}$
  - $S = [9+, 5-]$
  - $S_{Weak} = [6+, 2-]$
  - $S_{Strong} = [3+, 3-]$
- Information gain due to knowing *Wind*:

$$Gain(S, Wind) = Entropy(S) - 8/14\ Entropy(S_{Weak}) - 6/14\ Entropy(S_{Strong})$$

$$= 0.94 - 8/14 \times 0.811 - 6/14 \times 1.00$$

$$= 0.048$$

398

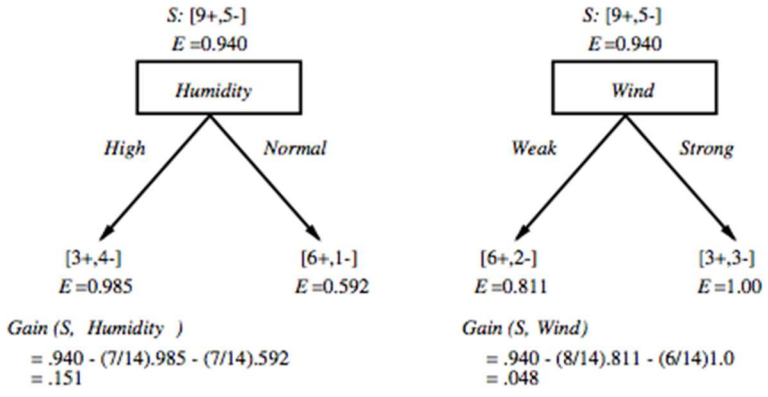
Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

399

Which attribute is the best classifier?

Which attribute is the best classifier?



Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

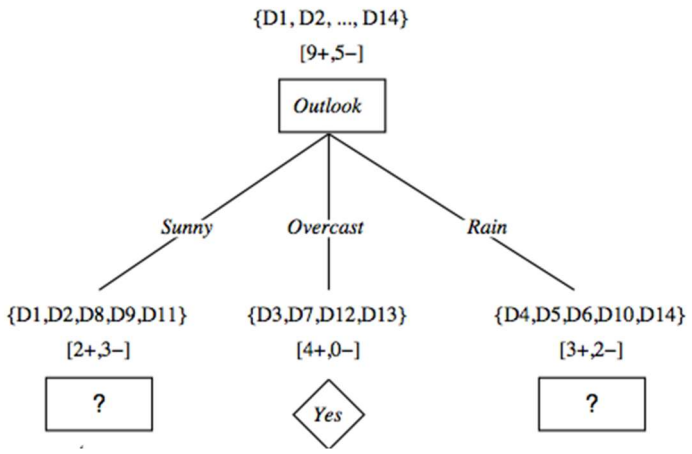
400

First step: which attribute to test at the root?

- Which attribute should be tested at the root?
  - $Gain(S, Outlook) = 0.246$
  - $Gain(S, Humidity) = 0.151$
  - $Gain(S, Wind) = 0.084$
  - $Gain(S, Temperature) = 0.029$
- *Outlook* provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of *Outlook*
  - partition the training samples according to the value of *Outlook*

401

After first step



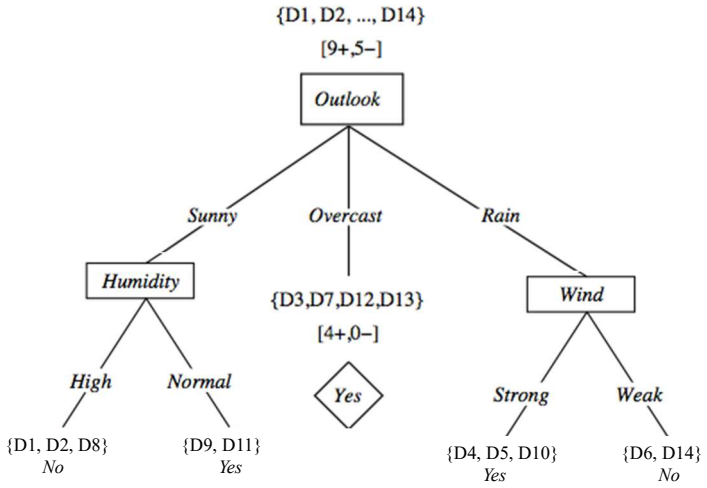
402

Second step

- Working on *Outlook=Sunny* node:
  $Gain(S_{Sunny}, Humidity) = 0.970 - 3/5 \times 0.0 - 2/5 \times 0.0 = 0.970$
  $Gain(S_{Sunny}, Wind) = 0.970 - 2/5 \times 1.0 - 3.5 \times 0.918 = 0\ .019$
  $Gain(S_{Sunny}, Temp.) = 0.970 - 2/5 \times 0.0 - 2/5 \times 1.0 - 1/5 \times 0.0 = 0.570$
- *Humidity* provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of *Humidity*
  - partition the training samples according to the value of *Humidity*

403

Second and third steps



404

ID3: algorithm

ID3(*X, T, Attrs*) *X*: training examples:
T: target attribute (e.g. *PlayTennis*),
Attrs: other attributes, initially all attributes

Create *Root* node
If all X's are +, *return Root* with class +
If all X's are –, *return Root* with class –
If Attrs is empty *return Root* with class most common value of *T* in *X*
else
  $A \leftarrow$ best attribute; decision attribute for Root $\leftarrow A$
  For each possible value v_i of *A:*
  - add a new branch below *Root,* for test $A = v_i$
  - $X_i \leftarrow$ subset of *X* with $A = v_i$
  - *If* X_i is empty *then* add a new leaf with class the most common value of *T* in *X*
    else add the subtree generated by ID3(X_i, *T, Attrs* – {*A*})
return Root

405

Prefer shorter hypotheses: Occam's razor

- Why prefer shorter hypotheses?
- Arguments in favor:
  - There are fewer short hypotheses than long ones
  - If a short hypothesis fits data unlikely to be a coincidence
  - Elegance and aesthetics
- Arguments against:
  - Not every short hypothesis is a reasonable one.
- Occam's razor says that when presented with competing hypotheses that make the **same** predictions, one should select the solution which is simple"

406

Issues in decision trees learning

- Overfitting
  - Reduced error pruning
  - Rule post-pruning
- Extensions
  - Continuous valued attributes
  - Handling training examples with missing attribute values

407

Overfitting: definition

- **overfitting** is "the production of an analysis that corresponds too closely or exactly to a particular set of data
- Building trees that “adapt too much” to the training examples may lead to “overfitting”.
- May therefore fail to fit additional data or predict future observations reliably
- **overfitted model** is a statistical model that contains more parameters than can be justified by the data

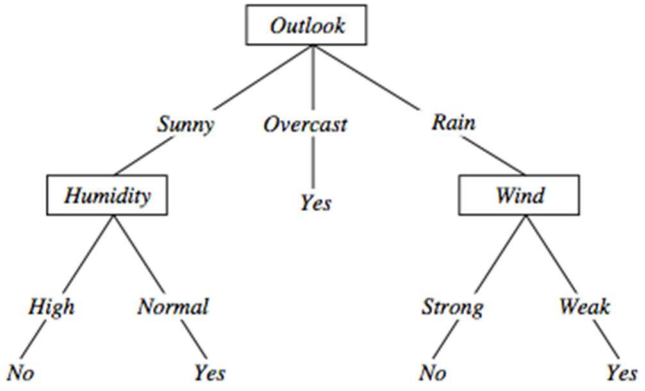
408

Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No
D15	Sunny	Hot	Normal	Strong	No

409

Overfitting in decision trees



⟨*Outlook=Sunny, Temp=Hot, Humidity=Normal, Wind=Strong, PlayTennis=No* ⟩

New noisy example causes splitting of second leaf node.

410

Avoid overfitting in Decision Trees

- Two strategies:
  1. Stop growing the tree earlier the tree, before perfect classification
  2. Allow the tree to *overfit* the data, and then *post-prune* the tree
- Training and validation set
  - split the training in two parts (training and validation) and use validation to assess the utility of *post-pruning*
    - *Reduced error pruning*
    - *Rule post pruning*

411

Reduced-error pruning

- Each node is a candidate for pruning
- *Pruning* consists in removing a subtree rooted in a node: the node becomes a leaf and is assigned the most common classification
- Nodes are removed only if the resulting tree performs no worse on the validation set.
- Nodes are pruned iteratively: at each iteration the node whose removal most increases accuracy on the validation set is pruned.
- Pruning stops when no pruning increases accuracy

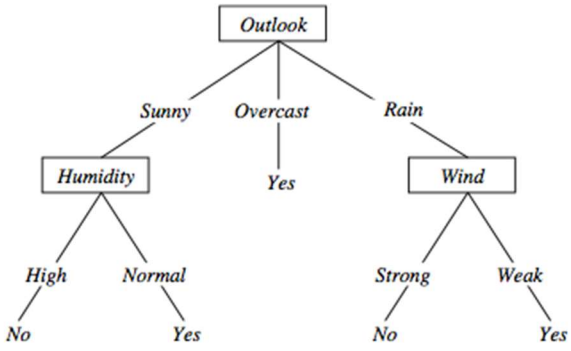
412

Rule post-pruning

1. Create the decision tree from the training set
2. Convert the tree into an equivalent set of rules
   - Each path corresponds to a rule
   - Each node along a path corresponds to a pre-condition
   - Each leaf classification to the post-condition
3. Prune (generalize) each rule by removing those preconditions whose removal improves accuracy …
   - … over validation set
4. Sort the rules in estimated order of accuracy, and consider them in sequence when classifying new instances

413

Converting to rules



$$(Outlook=Sunny) \wedge (Humidity=High) \Rightarrow (PlayTennis=No)$$

414

Rule Post-Pruning

- Convert tree to rules (one for each path from root to a leaf)
- For each antecedent in a rule, remove it if error rate on validation set does not decrease
- Sort final rule set by accuracy

```
Outlook=sunny ^ humidity=high -> No
Outlook=sunny ^ humidity=normal -> Yes
Outlook=overcast -> Yes
Outlook=rain ^ wind=strong -> No
Outlook=rain ^ wind=weak -> Yes
```

Compare first rule to:

```
Outlook=sunny->No
Humidity=high->No
```

Calculate accuracy of 3 rules based on validation set and pick best version.

415

Why converting to rules?

- Each distinct path produces a different rule: a condition removal may be based on a local (contextual) criterion. Node pruning is global and affects all the rules
- Provides flexibility of not removing entire node
- In rule form, tests are not ordered and there is no book-keeping involved when conditions (nodes) are removed
- Converting to rules improves readability for humans

416

Dealing with continuous-valued attributes

- Given a continuous-valued attribute A, dynamically create a new attribute A_c

  A_c = True *if* $A < c$, False *otherwise*
- How to determine threshold value c ?
- Example. *Temperature* in the *PlayTennis* example
  - Sort the examples according to *Temperature*

Temperature	40	48	|	60	72	80	|	90
PlayTennis	No	No	*54*	Yes	Yes	Yes	*85*	No

  - Determine candidate thresholds by averaging consecutive values where there is a change in classification: (48+60)/2=54 and (80+90)/2=85

417

Problems with information gain

- Natural bias of information gain: it favors attributes with many possible values.
- Consider the attribute *Date* in the *PlayTennis* example.
  - *Date* would have the highest information gain since it perfectly separates the training data.
  - It would be selected at the root resulting in a very broad tree
  - Very good on the training, this tree would perform poorly in predicting unknown instances. Overfitting.
- The problem is that the partition is too specific, too many small classes are generated.
- We need to look at alternative measures …

418

An alternative measure: gain ratio

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|}$$

- S_i are the sets obtained by partitioning on value i of A
- *SplitInformation* measures the entropy of S with respect to the values of A. The more uniformly dispersed the data the higher it is.

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInformation(S,A)}$$

- *GainRatio* penalizes attributes that split examples in many small classes such as *Date*. Let $|S|=n$, *Date* splits examples in n classes
  - $SplitInformation(S, Date) = -[(1/n \log_2 1/n) + \ldots + (1/n \log_2 1/n)] = -\log_2 1/n = \log_2 n$
- Compare with A, which splits data in two even classes:
  - $SplitInformation(S, A) = -[(1/2 \log_2 1/2) + (1/2 \log_2 1/2)] = -[-1/2 - 1/2] = 1$

419

Handling missing values training data

- How to cope with the problem that the value of some attribute may be missing?
- The strategy: use other examples to guess attribute
  1. Assign the value that is most common among the training examples at the node
  2. Assign a probability to each value, based on frequencies, and assign values to missing attribute, according to this probability distribution

420

Applications

Suited for following classification problems:

- Applications whose Instances are represented by attribute-value pairs.
- The target function has discrete output values
- Disjunctive descriptions may be required
- The training data may contain missing attribute values

Real world applications

- Biomedical applications
- Manufacturing
- Banking sector
- Make-Buy decisions

421

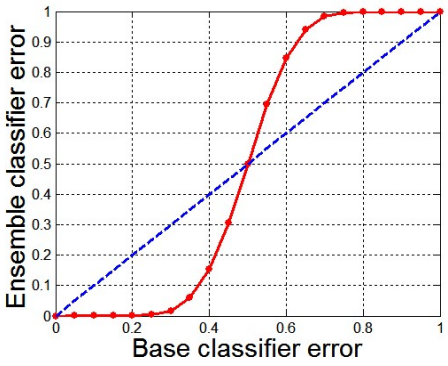
Ensemble Methods

- **Ensemble methods** use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone
- Construct a set of classifiers from the training data
- Predict class label of test records by combining the predictions made by multiple classifiers
- Tend to reduce problems related to over-fitting of the training data.

422

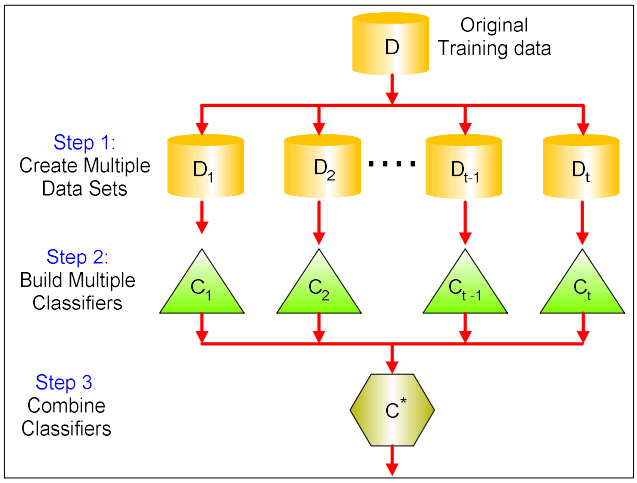
Why Ensemble Methods work?

- 25 base classifiers
- Each classifier has error rate, ε = 0.35
- If base classifiers are identical, then the ensemble will misclassify the same examples predicted incorrectly by the base classifiers depicted by dotted line
- Assume errors made by classifiers are uncorrelated
- ensemble makes a wrong prediction only if more than half of the base classifiers predict incorrectly
- Probability that the ensemble classifier makes a wrong prediction:



$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

General Approach



Simple Ensemble Techniques

innovate achieve lead

Max Voting

Ex: Movie rating

The result of max voting would be something like this:

- Rating by 5 friends: 5 4 5 4 4

Averaging- (5+4+5+4+4)/5 = 4.4 Final rating

Weighted Average

Weight-0.23 0.23 0.18 0.18 0.18

The result is calculated as [(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = **4.41.**

425

Types of Ensemble Methods

achieve lead

Manipulate data distribution

- Example: bagging

Manipulate input features

- Example: random forests

426

When does Ensemble work?

- Ensemble classifier performs better than the base classifiers when error is smaller than 0.5
- Necessary conditions for an ensemble classifier to perform better than a single classifier:
  - Base classifiers should be independent of each other
  - Base classifiers should do better than a classifier that performs random guessing

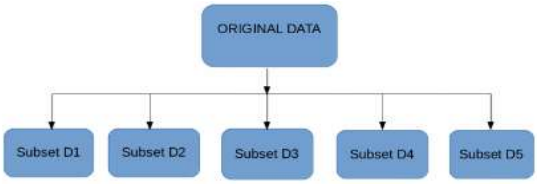
427

Bagging (Bootstrap Aggregating)

- Technique uses these subsets (bags) to get a fair idea of the distribution (complete set).
- The size of subsets created for bagging may be less than the original set.
- Bootstrapping is a sampling technique in which we create subsets of observations from the original dataset, **with replacement**.
- When you sample with replacement, items are independent. One item does not affect the outcome of the other. You have 1/7 chance of choosing the first item and a 1/7 chance of choosing the second item.
- If the two items are **dependent**, or linked to each other. When you choose the first item, you have a 1/7 probability of picking a item. Assuming you don't replace the item, you only have six items to pick from. That gives you a 1/6 chance of choosing a second item.

428

Bagging



- Multiple subsets are created from the original dataset, selecting observations with replacement.
- A base model (weak model) is created on each of these subsets.
- The models run in parallel and are independent of each other.
- The final predictions are determined by combining the predictions from all the models.

429

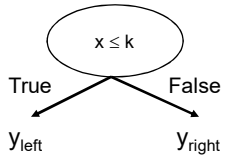
Bagging Example

- Consider 1-dimensional data set:

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

- Classifier is a decision stump
  - Decision rule: $x \leq k$ versus $x > k$
  - Split point k is chosen based on entropy



430

Bagging Example

innovate achieve lead

Bagging Round 1:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

x <= 0.35 → y = 1
x > 0.35 → y = -1

Bagging Round 2:

x	0.1	0.2	0.3	0.4	0.5	0.5	0.9	1	1	1
y	1	1	1	-1	-1	-1	1	1	1	1

x <= 0.7 → y = 1
x > 0.7 → y = 1

Bagging Round 3:

x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

x <= 0.35 → y = 1
x > 0.35 → y = -1

Bagging Round 4:

x	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

x <= 0.3 → y = 1
x > 0.3 → y = -1

Bagging Round 5:

x	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1
y	1	1	1	-1	-1	-1	-1	1	1	1

x <= 0.35 → y = 1
x > 0.35 → y = -1

431

Bagging Example

innovate achieve lead

Bagging Round 6:

x	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1
y	1	-1	-1	-1	-1	-1	-1	1	1	1

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 7:

x	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1
y	1	-1	-1	-1	-1	1	1	1	1	1

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 8:

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 9:

x	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

x <= 0.75 → y = -1
x > 0.75 → y = 1

Bagging Round 10:

x	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9
y	1	1	1	1	1	1	1	1	1	1

x <= 0.05 → y = 1
x > 0.05 → y = 1

432

Bagging Example

- Summary of Training sets:

Round	Split Point	Left Class	Right Class
1	0.35	1	-1
2	0.7	1	1
3	0.35	1	-1
4	0.3	1	-1
5	0.35	1	-1
6	0.75	-1	1
7	0.75	-1	1
8	0.75	-1	1
9	0.75	-1	1
10	0.05	1	1

Bagging Example

- Assume test set is the same as the original data
- Use majority vote to determine class of ensemble classifier

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Sum	2	2	2	-6	-6	-6	-6	2	2	2
Sign	1	1	1	-1	-1	-1	-1	1	1	1

Predicted Class

Bagging Algorithm

Algorithm 5.6 Bagging Algorithm

1: Let k be the number of bootstrap samples.
2: for $i = 1$ to k do
3: Create a bootstrap sample of size n, D_i.
4: Train a base classifier C_i on the bootstrap sample D_i.
5: end for
6: $C^*(x) = \arg\max_y \sum_i \delta\big(C_i(x) = y\big)$, $\{\delta(\cdot) = 1$ if its argument is true, and 0 otherwise.$\}$

435

Random Forest

- Random Forest is another ensemble machine learning algorithm that follows the bagging technique.
- The base estimators in random forest are decision trees.
- Random forest randomly selects a set of features which are used to decide the best split at each node of the decision tree.

436

Random Forest

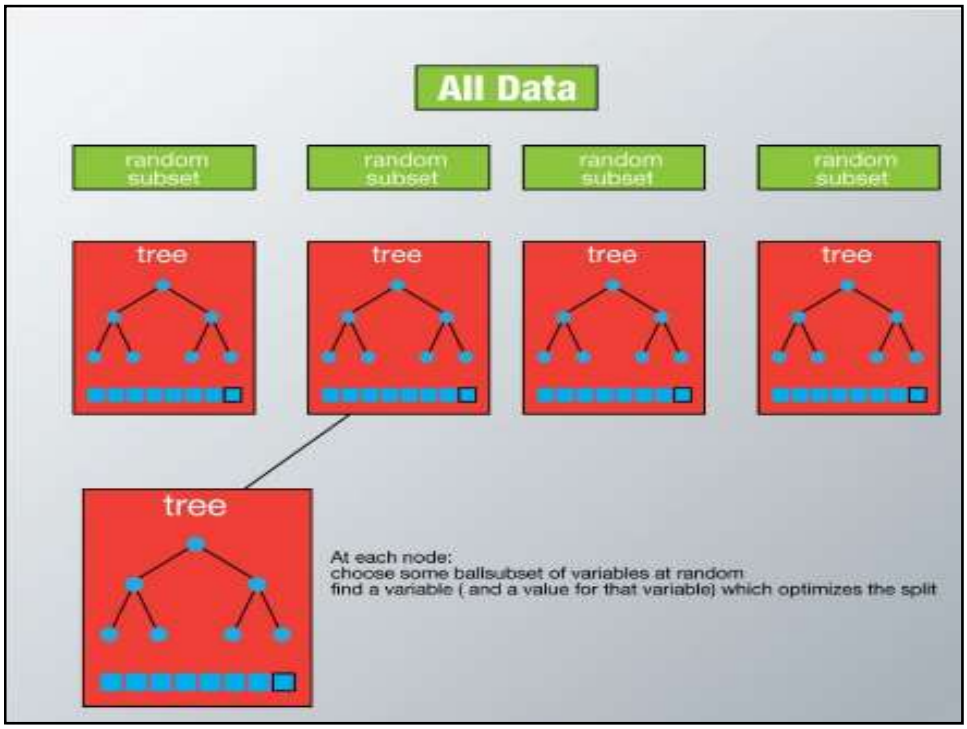
- As in bagging, we build a number of decision trees on bootstrapped training samples each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors.
- Note that if m = p, then this is bagging.

437

Random Forest

- Random subsets are created from the original dataset (bootstrapping).
- At each node in the decision tree, only a random set of features are considered to decide the best split.
- A decision tree model is fitted on each of the subsets.
- The final prediction is calculated by averaging the predictions from all decision trees.

438



439

Random Forests Algorithm

innovate achieve lead

- For b = 1 to B:
  - (a) Draw a bootstrap sample Z∗ of size N from the training data.
  - (b) Grow a random-forest tree to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size nmin is reached.
    - i. Select m variables at random from the p variables.
    - ii. Pick the best variable/split-point among the m.
    - iii. Split the node into two daughter nodes. Output the ensemble of trees.
- To make a prediction at a new point x we do:
  - For regression: average the results
  - For classification: majority vote

440

Random Forests Tuning

- The inventors make the following recommendations:
  - For classification, the default value for m is √p and the minimum node size is one.
  - For regression, the default value for m is p/3 and the minimum node size is five.
- In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters.

441

Example

- 4,718 genes measured on tissue samples from 349 patients.
- Each gene has different expression
- Each of the patient samples has a qualitative label with 15 different levels: either normal or 1 of 14 different types of cancer.
- Use random forests to predict cancer type based on the 500 genes that have the largest variance in the training set.

442

Random Forests Issues

- When the number of variables is large, but the fraction of relevant variables is small, random forests are likely to perform poorly when m is small

Why?

- Because: At each split the chance can be small that the relevant variables will be selected
- For example, with 3 relevant and 100 not so relevant variables the probability of any of the relevant variables being selected at any split is ~0.25

Advantages of Random Forest

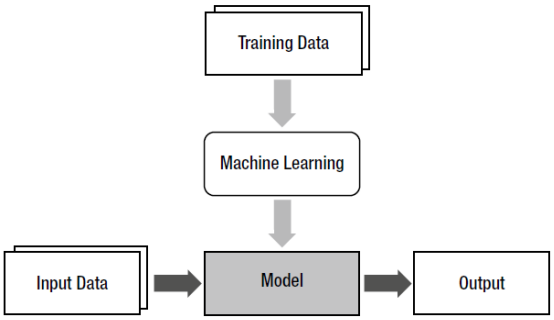
- Algorithm can solve both type of problems i.e. classification and regression
- Power of handle large data set with higher dimensionality.
- It can handle thousands of input variables and identify most significant variables so it is considered as one of the dimensionality reduction methods.
- Model outputs **Importance of variable,** which can be a very handy feature (on some random data set).

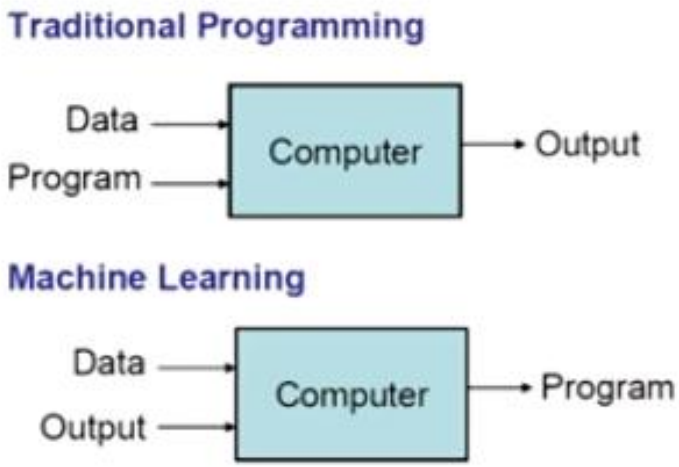
Disadvantages of Random Forest

- May over-fit data sets that are particularly noisy.
- Random Forest can feel like a black box approach for statistical modelers – you have very little control on what the model does. You can at best – try different parameters and random seeds!

Machine Learning and Examples of its Applications

Machine Learning is a modeling technique that involves data. **Machine learning** is defined as an automated process that extracts patterns from data. To build the models used in predictive data analytics applications, we use **supervised machine learning**.





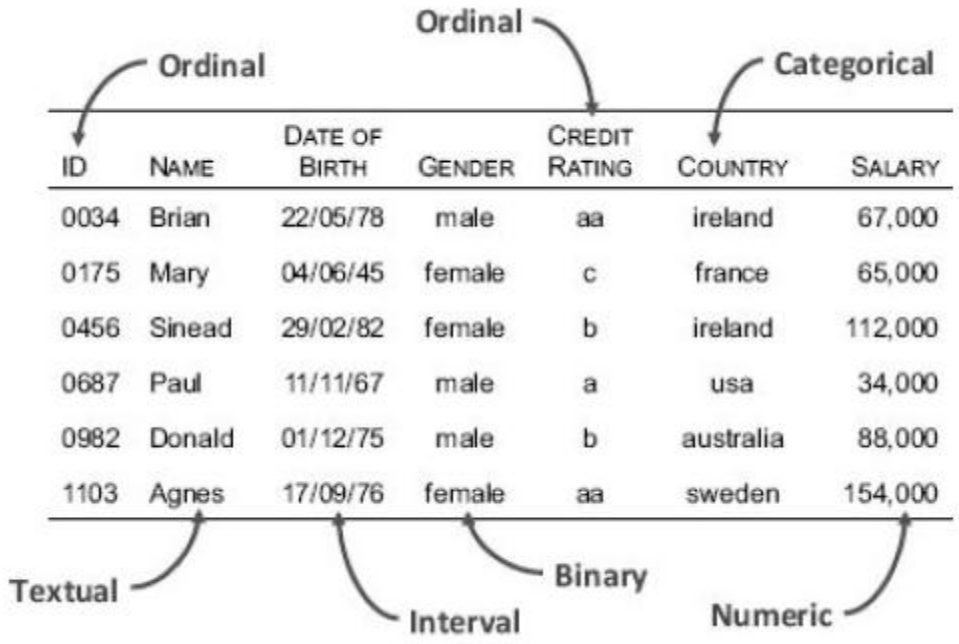
A computer program is said to learn from experience**E**with respect to some class of tasks**T**and performance measure**P**, if its performance at tasks in **T**, as measured by , improves with experience **E**. (Tom Mitchell, 1998)

A computer program that learns to play checkers might improve its performance as ***measured by its ability to win***at the class of tasks involving ***playing checkers games,*** through experience ***obtained*** by ***playing games against itself***

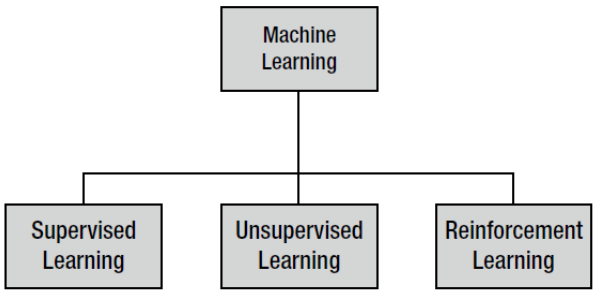
Task**T**: playing checkers
•Performance measure**P**: % of game won against opponents
•Training experience**E**: playing practice game against itself

Types of Data:
Numeric: True numeric values that allow arithmetic operations (e.g., price, age)

Interval: Values that allow ordering and subtraction, but do not allow other arithmetic operations (e.g., date, time)
Ordinal: Values that allow ordering but do not permit arithmetic (e.g., size measured as small, medium, or large)
Categorical: A finite set of values that cannot be ordered and allow no arithmetic (e.g., country, product type)
Binary: A set of just two values (e.g., gender)
Textual: Free-form, usually short, text data (e.g., name, address)



Types of Machine Learning
Many different types of Machine Learning techniques have been developed to solve problems in various fields. These Machine Learning techniques can be classified into three types depending on the training method (see Figure).
• Supervised learning
• Unsupervised learning
• Reinforcement learning



ML tools:
Regression
– predict new values based on the past, inference

– compute the new values for a dependent variable based on the values of one or more measured attributes

• Classification:

– divide samples in classes

– use a trained set of previously labeled data

• Clustering

– partitioning of a data set into subsets (clusters) so that data in each subset ideally share some common characteristics

Classification

There are two ways to assign a new value to a given class.

• **Crispy classification**

– given an input, the classifier returns its label

• **Probabilistic classification**

– given an input, the classifier returns its probabilities to belong to each class

– useful when some mistakes can be more costly than others (give me only data >90%)

Training Method	Training Data
Supervised Learning	{ input, correct output }
Unsupervised Learning	{ input }
Reinforced Learning	{ input, some output, grade for this output }

Application areas:

Speech and Hand Writing Recognition)

•Robotics (Robot locomotion)

•Search Engines (Information Retrieval)

•Learning to Classify new astronomical structures

•Medical Diagnosis

•Learning to drive an autonomous vehicle

•Computational Biology/Bioinformatics

•Computer Vision (Object Detection algorithms)

•Detecting credit card fraud

•Stock Market analysis

•Game playing

Possible Issues in ML:

- What algorithms are available for learning a concept? How well do they perform?
- How much training data is sufficient to learn a concept with high confidence?
- When is it useful to use prior knowledge?
- Are some training examples more useful than others?
- What are the best tasks for a system to learn?

- What is the best way for a system to represent its knowledge?

Designing a Learning system- Training Experience

- One key attribute is whether the training experience provides ***direct*** or ***indirect*** feedback regarding the choices made by the performance system.
- A second important attribute of the training experience is the degree to which the learner controls the sequence of training examples[supervise/unsupervised].
- A third important attribute of the training experience is how well it represents the distribution of examples over which the final system performance ***P*** must be measured[train/test].
- Move to numerical domain (or) assign values, V: S-> R.
- More expressive the function, the closer it is to the truth but will need more training examples[regression equation].

Linear Algebra:

Types of matrices:

Diagonal Matrix D. Scalar Matrix S. Unit Matrix I

$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & c \end{bmatrix}, \quad \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Upper and Lower Triangular Matrices

$$\begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 4 & 2 \\ 0 & 3 & 2 \\ 0 & 0 & 6 \end{bmatrix}, \quad \begin{bmatrix} 2 & 0 & 0 \\ 8 & -1 & 0 \\ 7 & 6 & 8 \end{bmatrix}, \quad \begin{bmatrix} 3 & 0 & 0 & 0 \\ 9 & -3 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 9 & 3 & 6 \end{bmatrix}.$$

Upper triangular Lower triangular

Symmetric, Skew-Symmetric, and Orthogonal Matrices

A *real* square matrix $\mathbf{A} = [a_{jk}]$ is called

symmetric if transposition leaves it unchanged,

(1) $$\mathbf{A}^T = \mathbf{A}, \qquad \text{thus} \qquad a_{kj} = a_{jk},$$

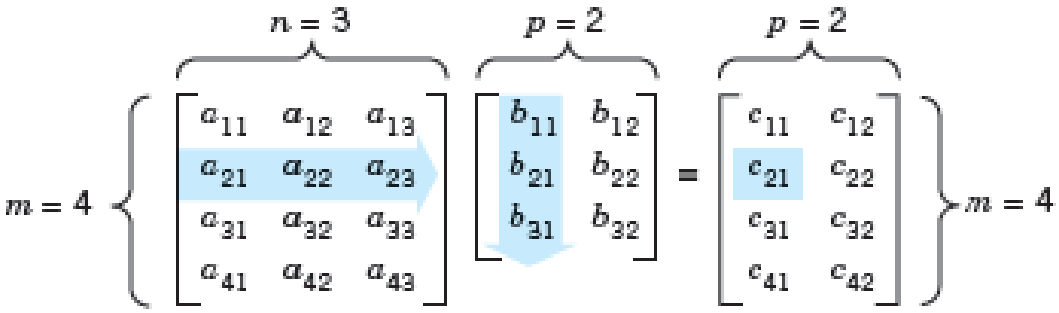
skew-symmetric if transposition gives the negative of $\mathbf{A}$,

(2) $$\mathbf{A}^T = -\mathbf{A}, \qquad \text{thus} \qquad a_{kj} = -a_{jk},$$

orthogonal if transposition gives the inverse of $\mathbf{A}$,

(3) $$\mathbf{A}^T = \mathbf{A}^{-1}.$$

Operations on matrices:



Notations in a product **AB = C**

Matrix Multiplication

$$\mathbf{AB} = \begin{bmatrix} 3 & 5 & -1 \\ 4 & 0 & 2 \\ -6 & -3 & 2 \end{bmatrix} \begin{bmatrix} 2 & -2 & 3 & 1 \\ 5 & 0 & 7 & 8 \\ 9 & -4 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 22 & -2 & 43 & 42 \\ 26 & -16 & 14 & 6 \\ -9 & 4 & -37 & -28 \end{bmatrix}$$

Here $c_{11} = 3 \cdot 2 + 5 \cdot 5 + (-1) \cdot 9 = 22$, and so on. The entry in the box is $c_{23} = 4 \cdot 3 + 0 \cdot 7 + 2 \cdot 1 = 14$.
The product BA is not defined. ■

Product is not commutative:

$$\begin{bmatrix} 1 & 1 \\ 100 & 100 \end{bmatrix}\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{but} \quad \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 100 & 100 \end{bmatrix} = \begin{bmatrix} 99 & 99 \\ -99 & -99 \end{bmatrix}.$$

(a) $(k\mathbf{A})\mathbf{B} = k(\mathbf{AB}) = \mathbf{A}(k\mathbf{B})$ *written* $k\mathbf{AB}$ *or* $\mathbf{A}k\mathbf{B}$

(b) $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$ *written* $\mathbf{ABC}$

(c) $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$

(d) $\mathbf{C}(\mathbf{A} + \mathbf{B}) = \mathbf{CA} + \mathbf{CB}$

Transposition:

$$\mathbf{A} = \begin{bmatrix} 5 & -8 & 1 \\ 4 & 0 & 0 \end{bmatrix}, \quad \text{then} \quad \mathbf{A}^T = \begin{bmatrix} 5 & 4 \\ -8 & 0 \\ 1 & 0 \end{bmatrix}.$$

(a) $(\mathbf{A}^T)^T = \mathbf{A}$

(b) $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$

(c) $(c\mathbf{A})^T = c\mathbf{A}^T$

(d) $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T.$

Ex:

- ☐ Sales figures for three products I, II, III in a store on Monday (Mon), Tuesday (Tues), may for each week…be arranged in a matrix

$$A = \begin{array}{c} \begin{matrix} \text{Mon} & \text{Tues} & \text{Wed} & \text{Thur} & \text{Fri} & \text{Sat} & \text{Sun} \end{matrix} \\ \begin{bmatrix} 40 & 33 & 81 & 0 & 21 & 47 & 33 \\ 0 & 12 & 78 & 50 & 50 & 96 & 90 \\ 10 & 0 & 0 & 27 & 43 & 78 & 56 \end{bmatrix} \end{array} \begin{matrix} \text{I} \\ \cdot \text{ II} \\ \text{III} \end{matrix}$$

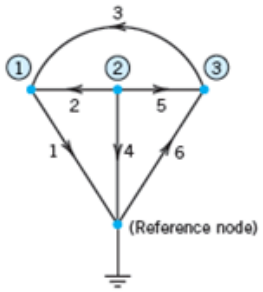
Ex2:

- **Nodal Incidence Matrix. The network in picture** consists of six *branches (connections) and four nodes* (points where two or more branches come together). One node is the *reference node (grounded node, whose* voltage is zero). We number the other nodes and number and direct the branches. This we do arbitrarily. The network can now be described by a matrix. **A is called the *nodal incidence matrix of the network.*** Show that for the network in Fig. the matrix **A has** the given form.

Nodal Incidence Matrix.

$\mathbf{A} = [a_{jk}]$, where

$$a_{jk} = \begin{cases} +1 \text{ if branch } k \text{ leaves node } (j) \\ -1 \text{ if branch } k \text{ enters node } (j) \\ 0 \text{ if branch } k \text{ does not touch node } (j). \end{cases}$$



Branch	1	2	3	4	5	6
Node ①	1	−1	−1	0	0	0
Node ②	0	1	0	1	1	0
Node ③	0	0	1	0	−1	−1

Fig. 155. Network and nodal incidence

Special Matrices

- Any Matrix can be written as a sum of a symmetric and skew-symmetric matrices

$$\mathbf{R} = \tfrac{1}{2}(\mathbf{A} + \mathbf{A}^{\mathsf{T}}) \quad \text{and} \quad \mathbf{S} = \tfrac{1}{2}(\mathbf{A} - \mathbf{A}^{\mathsf{T}}).$$

$$\mathbf{A} = \begin{bmatrix} 9 & 5 & 2 \\ 2 & 3 & -8 \\ 5 & 4 & 3 \end{bmatrix} = \mathbf{R} + \mathbf{S} = \begin{bmatrix} 9.0 & 3.5 & 3.5 \\ 3.5 & 3.0 & -2.0 \\ 3.5 & -2.0 & 3.0 \end{bmatrix} + \begin{bmatrix} 0 & 1.5 & -1.5 \\ -1.5 & 0 & -6.0 \\ 1.5 & 6.0 & 0 \end{bmatrix}$$

Matrix multiplication:

- Supercomp Ltd produces two computer models PC1086 and PC1186. The matrix **A shows the cost per computer**(in thousands of dollars) and **B the production figures for the year 2010 (in multiples of 10,000 units.) Find a**matrix **C that shows the shareholders the cost per quarter (in millions of dollars) for raw material, labor, and**miscellaneous.

$$\mathbf{A} = \begin{array}{c} \begin{matrix} \text{PC1086} & \text{PC1186} \end{matrix} \\ \begin{bmatrix} 1.2 & 1.6 \\ 0.3 & 0.4 \\ 0.5 & 0.6 \end{bmatrix} \end{array} \begin{matrix} \\ \text{Raw Components} \\ \text{Labor} \\ \text{Miscellaneous} \end{matrix}$$

$$\mathbf{B} = \begin{array}{c} \begin{matrix} \text{Quarter} \\ 1 \quad 2 \quad 3 \quad 4 \end{matrix} \\ \begin{bmatrix} 3 & 8 & 6 & 9 \\ 6 & 2 & 4 & 3 \end{bmatrix} \end{array} \begin{matrix} \\ \text{PC1086} \\ \text{PC1186} \end{matrix}$$

$$\mathbf{C} = \mathbf{AB} = \begin{array}{c} \begin{matrix} \text{Quarter} \\ 1 \quad 2 \quad 3 \quad 4 \end{matrix} \\ \begin{bmatrix} 13.2 & 12.8 & 13.6 & 15.6 \\ 3.3 & 3.2 & 3.4 & 3.9 \\ 5.1 & 5.2 & 5.4 & 6.3 \end{bmatrix} \end{array} \begin{matrix} \\ \text{Raw Components} \\ \text{Labor} \\ \text{Miscellaneous} \end{matrix}$$

- Since cost is given in multiples of 1000and production in multiples of 10,000 units, the entries of **C are**
- multiples of 10millions; thus c11=132means 132 million, etc.

Cramer's rule:

Cramer's Theorem (Solution of Linear Systems by Determinants)

(a) *If a linear system of n equations in the same number of unknowns* $x_1, \cdots, x_n$

$$\text{(6)} \qquad \begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

has a nonzero coefficient determinant $D = \det \mathbf{A}$, *the system has precisely one solution. This solution is given by the formulas*

$$\text{(7)} \qquad x_1 = \frac{D_1}{D}, \quad x_2 = \frac{D_2}{D}, \cdots, \quad x_n = \frac{D_n}{D} \qquad \textbf{(Cramer's rule)}$$

where D_k *is the determinant obtained from D by replacing in D the kth column by the column with the entries* $b_1, \cdots, b_n$.

(b) *Hence if the system* (6) *is* ***homogeneous*** *and* $D \neq 0$, *it has only the trivial solution* $x_1 = 0, x_2 = 0, \cdots, x_n = 0$. *If* $D = 0$, *the homogeneous system also has nontrivial solutions.*

Linear system of equations:

- ☐ Existence of solution: Coefficient matrix A and Augmented matrix have the same rank.
- ☐ Uniqueness: If the common rank is n, the order of the matrix.
- ☐ If common rank r<n, infinite solutions exist.
- ☐ Now we can use methods like Gauss-Elimination to obtain them.

Limits:

consider the functions

$$f(x) = \frac{x^2 - 4}{x - 2} \quad \text{and} \quad g(x) = \frac{x^2 - 5}{x - 2}.$$

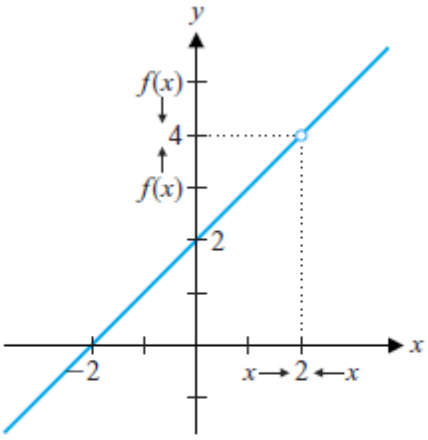


FIGURE 1.7a

$$y = \frac{x^2 - 4}{x - 2}$$

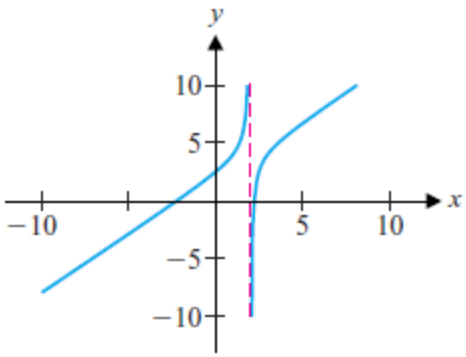


FIGURE 1.7b

$$y = \frac{x^2 - 5}{x - 2}$$

x	$f(x) = \dfrac{x^2 - 4}{x - 2}$
1.9	3.9
1.99	3.99
1.999	3.999
1.9999	3.9999

x	$f(x) = \dfrac{x^2 - 4}{x - 2}$
2.1	4.1
2.01	4.01
2.001	4.001
2.0001	4.0001

Notice that as you move down the first column of the table, the x-values get closer to 2, but are all less than 2. We use the notation $x \to 2^-$ to indicate that x *approaches 2 from the left side*. Notice that the table and the graph both suggest that as x gets closer and closer to 2 (with $x < 2$), $f(x)$ is getting closer and closer to 4. In view of this, we say that the **limit of $f(x)$ as x approaches 2 from the left** is 4, written

$$\lim_{x \to 2^-} f(x) = 4.$$

Likewise, we need to consider what happens to the function values for x close to 2 but larger than 2. Here, we use the notation $x \to 2^+$ to indicate that x *approaches 2 from the right side*. We compute some of these values in the second table.

Again, the table and graph both suggest that as x gets closer and closer to 2 (with $x > 2$), $f(x)$ is getting closer and closer to 4. In view of this, we say that the **limit of $f(x)$ as x approaches 2 from the right** is 4, written

$$\lim_{x \to 2^+} f(x) = 4.$$

We call $\lim_{x \to 2^-} f(x)$ and $\lim_{x \to 2^+} f(x)$ **one-sided limits.** Since the two one-sided limits of $f(x)$ are the same, we summarize our results by saying that the **limit of $f(x)$ as x approaches 2** is 4, written

$$\lim_{x \to 2} f(x) = 4.$$

The notion of limit as we have described it here is intended to communicate the behavior of a function *near* some point of interest, but not actually *at* that point. We finally observe that we can also determine this limit algebraically, as follows. Notice that since the expression in the numerator of $f(x) = \dfrac{x^2 - 4}{x - 2}$ factors, we can write

$$\begin{aligned}
\lim_{x \to 2} f(x) &= \lim_{x \to 2} \frac{x^2 - 4}{x - 2} \\
&= \lim_{x \to 2} \frac{(x - 2)(x + 2)}{x - 2} && \text{Cancel the factors of } (x - 2). \\
&= \lim_{x \to 2} (x + 2) = 4, && \text{As } x \text{ approaches 2, } (x + 2) \text{ approaches 4.}
\end{aligned}$$

where we can cancel the factors of $(x - 2)$ since in the limit as $x \to 2$, x is *close* to 2, but $x \neq 2$, so that $x - 2 \neq 0$.

A limit exists if and only if both corresponding one-sided limits exist and are equal. That is,

$$\lim_{x \to a} f(x) = L, \text{ for some number } L, \text{ if and only if } \lim_{x \to a^-} f(x) = \lim_{x \to a^+} f(x) = L.$$

THEOREM 3.1

Suppose that $\lim_{x \to a} f(x)$ and $\lim_{x \to a} g(x)$ both exist and let c be any constant. The following then apply:

(i) $\lim_{x \to a} [c \cdot f(x)] = c \cdot \lim_{x \to a} f(x)$,

(ii) $\lim_{x \to a} [f(x) \pm g(x)] = \lim_{x \to a} f(x) \pm \lim_{x \to a} g(x)$,

(iii) $\lim_{x \to a} [f(x) \cdot g(x)] = \left[\lim_{x \to a} f(x)\right]\left[\lim_{x \to a} g(x)\right]$ and

(iv) $\lim_{x \to a} \dfrac{f(x)}{g(x)} = \dfrac{\lim_{x \to a} f(x)}{\lim_{x \to a} g(x)} \left(\text{if } \lim_{x \to a} g(x) \neq 0\right)$.

EXAMPLE 3.1 Finding the Limit of a Polynomial

Apply the rules of limits to evaluate $\lim_{x\to 2}(3x^2 - 5x + 4)$.

Solution We have

$$\begin{aligned}
\lim_{x\to 2}(3x^2 - 5x + 4) &= \lim_{x\to 2}(3x^2) - \lim_{x\to 2}(5x) + \lim_{x\to 2} 4 && \text{By Theorem 3.1 (ii).}\\
&= 3\lim_{x\to 2} x^2 - 5\lim_{x\to 2} x + 4 && \text{By Theorem 3.1 (i).}\\
&= 3\cdot(2)^2 - 5\cdot 2 + 4 = 6. && \text{By (3.4).}
\end{aligned}$$

EXAMPLE 3.2 Finding the Limit of a Rational Function

Apply the rules of limits to evaluate $\lim_{x\to 3}\dfrac{x^3 - 5x + 4}{x^2 - 2}$.

Solution We get

$$\begin{aligned}
\lim_{x\to 3}\frac{x^3 - 5x + 4}{x^2 - 2} &= \frac{\lim_{x\to 3}(x^3 - 5x + 4)}{\lim_{x\to 3}(x^2 - 2)} && \text{By Theorem 3.1 (iv).}\\
&= \frac{\lim_{x\to 3} x^3 - 5\lim_{x\to 3} x + \lim_{x\to 3} 4}{\lim_{x\to 3} x^2 - \lim_{x\to 3} 2} && \text{By Theorem 3.1 (i) and (ii).}\\
&= \frac{3^3 - 5\cdot 3 + 4}{3^2 - 2} = \frac{16}{7}. && \text{By (3.4).}
\end{aligned}$$

Continuity:

When you describe something as *continuous,* just what do you have in mind? For example, if told that a machine has been in *continuous* operation for the past 60 hours, most of us would interpret this to mean that the machine has been in operation *all* of that time, without any interruption at all, even for a moment. Mathematicians mean much the same thing when we say that a function is continuous. A function is said to be *continuous* on an interval if its graph on that interval can be drawn without interruption, that is, without lifting your pencil from the paper.

It is helpful for us to first try to see what it is about the functions whose graphs are shown in Figures 1.22a–1.22d that makes them *discontinuous* (i.e., not continuous) at the point $x = a$.

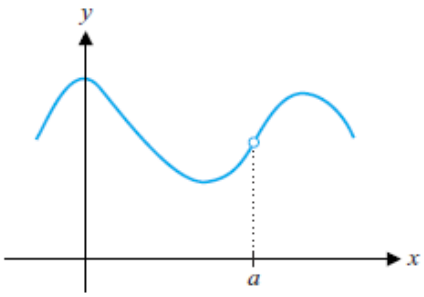


FIGURE 1.22a
$f(a)$ is not defined (the graph has a hole at $x = a$).

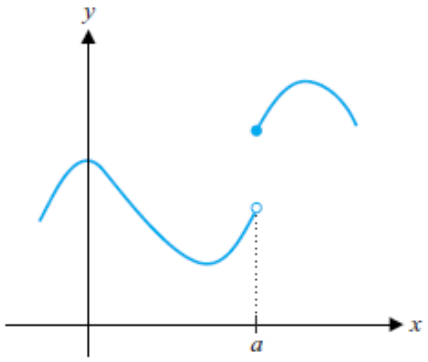


FIGURE 1.22b
$f(a)$ is defined, but $\lim_{x \to a} f(x)$ does not exist (the graph has a jump at $x = a$).

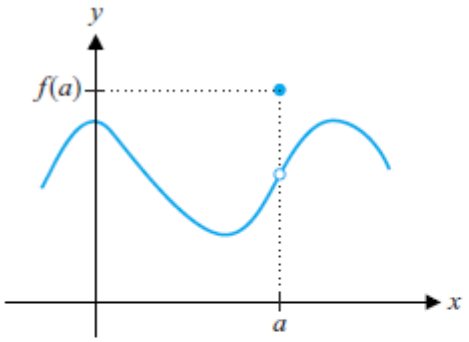


FIGURE 1.22c
$\lim_{x \to a} f(x)$ exists and $f(a)$ is defined, but $\lim_{x \to a} f(x) \neq f(a)$ (the graph has a hole at $x = a$).

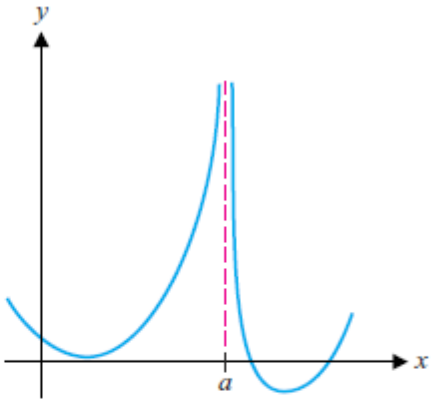


FIGURE 1.22d
$\lim_{x \to a} f(x)$ does not exist (the function "blows up" at $x = a$).

DEFINITION 4.1

A function f is **continuous** at $x = a$ when
(i) $f(a)$ is defined, (ii) $\lim_{x \to a} f(x)$ exists and (iii) $\lim_{x \to a} f(x) = f(a)$.
Otherwise, f is said to be **discontinuous** at $x = a$.

Determine where $f(x) = \dfrac{x^2 + 2x - 3}{x - 1}$ is continuous.

Solution Note that

$$f(x) = \frac{x^2 + 2x - 3}{x - 1} = \frac{(x - 1)(x + 3)}{x - 1} \qquad \text{Factoring the numerator.}$$

$$= x + 3, \text{ for } x \neq 1. \qquad \text{Canceling common factors.}$$

This says that the graph of f is a straight line, but with a hole in it at $x = 1$, as indicated in Figure 1.23. So, f is discontinuous at $x = 1$, but continuous elsewhere. ■

THEOREM 4.2

Suppose that f and g are continuous at $x = a$. Then all of the following are true:

(i) $(f \pm g)$ is continuous at $x = a$,
(ii) $(f \cdot g)$ is continuous at $x = a$ and
(iii) (f/g) is continuous at $x = a$ if $g(a) \neq 0$.

DEFINITION 4.2

If f is continuous at every point on an open interval (a, b), we say that f is **continuous on** (a, b). Following Figure 1.27, we say that f is **continuous on the closed interval** $[a, b]$, if f is continuous on the open interval (a, b) and

$$\lim_{x \to a^+} f(x) = f(a) \quad \text{and} \quad \lim_{x \to b^-} f(x) = f(b).$$

Finally, if f is continuous on all of $(-\infty, \infty)$, we simply say that f is **continuous.** (That is, when we don't specify an interval, we mean continuous everywhere.)

1. Partial Derivatives

The derivative of a function of one variable, such as $y(x)$, tells us the gradient of the function: how y changes when x increases. If we have a function of more than one variable, such as:

$$z(x, y) = x^3 + 4xy + 5y^2$$

we can ask, for example, how z changes when x increases but y doesn't change. The answer to this question is found by thinking of z as a function of x, and differentiating, treating y as if it were a constant parameter:

$$\frac{\partial z}{\partial x} = 3x^2 + 4y$$

This process is called partial differentiation. We write $\frac{\partial z}{\partial x}$ rather than $\frac{dz}{dx}$, to emphasize that z is a function of another variable as well as x, which is being held constant.

$\frac{\partial z}{\partial x}$ is called the partial derivative of z with respect to x

EXERCISES 7.1: Find the partial derivatives with respect to x and y of the functions:

(1) $f(x, y) = 3x^2 - xy^4$

(2) $h(x, y) = (x + 1)^2(y + 2)$

(3) $g(x, y) = \dfrac{\ln x}{y}$

1.1. Second-order Partial Derivatives

For the function in the previous section:

$$z(x, y) = x^3 + 4xy + 5y^2$$

we found:

$$\frac{\partial z}{\partial x} = 3x^2 + 4y$$
$$\frac{\partial z}{\partial y} = 4x + 10y$$

These are the *first-order* partial derivatives. But we can differentiate again to find *second-order* partial derivatives. The second derivative with respect to x tells us how $\frac{\partial z}{\partial x}$ changes as x increases, still keeping y constant.

$$\frac{\partial^2 z}{\partial x^2} = 6x$$

Similarly:

$$\frac{\partial^2 z}{\partial y^2} = 10$$

2. Economic Applications of Partial Derivatives, and Euler's Theorem

2.1. The Marginal Products of Labour and Capital

Suppose that the output produced by a firm depends on the amounts of labour and capital used. If the production function is

$$Y(K, L)$$

the partial derivative of Y with respect to L tells us the the marginal product of labour:

$$MPL = \frac{\partial Y}{\partial L}$$

The marginal product of labour is the amount of extra output the firm could produce if it used one extra unit of labour, but kept capital the same as before.
Similarly the marginal product of capital is:

$$MPK = \frac{\partial Y}{\partial K}$$

EXAMPLES 2.1: For a firm with production function $Y(K, L) = 5K^{\frac{1}{3}}L^{\frac{2}{3}}$:

(i) Find the marginal product of labour.

$$MPL = \frac{\partial Y}{\partial L} = \frac{10}{3}K^{\frac{1}{3}}L^{-\frac{1}{3}}$$

(ii) What is the MPL when $K = 64$ and $L = 125$?

$$MPL = \frac{10}{3}(64)^{\frac{1}{3}}(125)^{-\frac{1}{3}} = \frac{10}{3} \times 4 \times \frac{1}{5} = \frac{8}{3}$$

(iii) What happens to the marginal product of labour as the number of workers increases?

Differentiate MPL with respect to L: $\dfrac{\partial^2 Y}{\partial L^2} = -\dfrac{10}{9}K^{\frac{1}{3}}L^{-\frac{4}{3}} < 0$

So the MPL decreases as the labour input increases – the firm has diminishing returns to labour, if capital is held constant. This is true for *all* values of K and L.

5. The Chain Rule and Implicit Differentiation

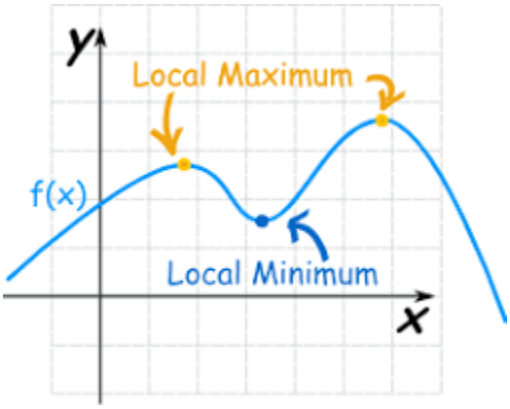
5.1. The Chain Rule for Functions of Several Variables

If z is a function of two variables, x and y, and both x and y depend on another variable, t (time, for example), then z also depends on t. We have:

If $z = z(x, y)$, and x and y are functions of t, then:

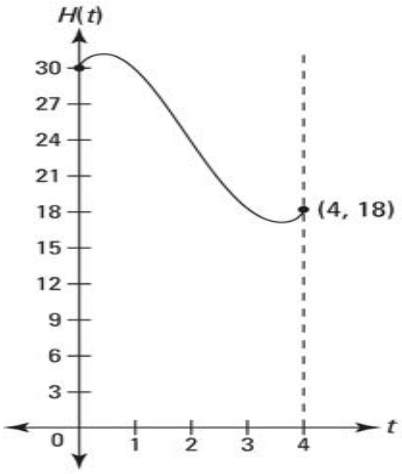
$$\frac{dz}{dt} = \frac{\partial z}{\partial x}\frac{dx}{dt} + \frac{\partial z}{\partial y}\frac{dy}{dt}$$

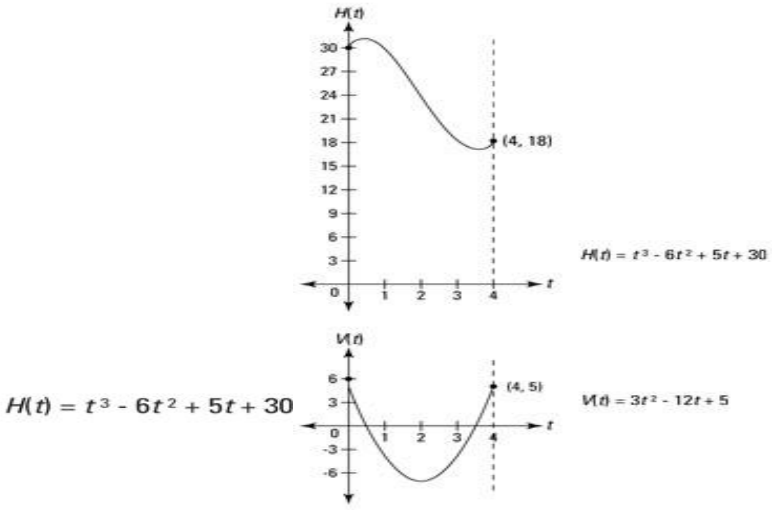
Maxima-Minima:



Procedure:
Step1: Find the first derivative of f(x), equate it to zero and solve.
Step2: Substitute the points obtained in step 1 on the second derivative. If value is –ve, then the point is max and if +ve, then the point is minimum.

Example: A yo-yo moves straight up and down. Its height above the ground, as a function of time, is given by the function below, where t is in seconds and $H(t)$ is in inches. At $t = 0$, it's 30 inches above the ground, and after 4 seconds, it's at height of 18 inches.





- **Velocity,** $V(t)$ is the derivative of position (height, in this problem) and acceleration, $A(t)$, is the derivative of velocity. Thus $V(t)=3t^2-12t+5$
-
- **Maximum and minimum height** of $H(t)$ occur at the local extrema you see in the above figure. To locate them, set the derivative of $H(t)$ — that's $V(t)$ — equal to zero and solve.
- These are the times when the yo-yo reaches its maximum and minimum heights. Plug these numbers into $H(t)$ to obtain the heights:
- $H(0.47) \approx 31.1$
- $H(3.53) \approx 16.9$
- So, the yo-yo gets as high as about 31.1 inches above the ground at $t \approx 0.47$ seconds and as low as about 16.9 inches at $t \approx 3.53$ seconds.
- **Maximum and minimum velocity** of the yo-yo during the interval from 0 to 4 seconds are determined with the derivative of $V(t)$: Set the derivative of $V(t)$ — that's $A(t)$ — equal to zero and solve:
- $V'(t)=A(t)$. Hence $A(t)=6t-12$, which when equated to zero, we obtain $t=2$.
- Now, evaluate $V(t)$ at the critical number, 2, and at the interval's endpoints, 0 and 4:

Decision theory

- Broad types are
- Normative and Descriptive:
- A normative decision theory is a theory about how decisions should be made, and a descriptive theory is a theory about how decisions are actually made.
- Decision process:
- 1.Identification of the problem
- 2. Obtaining necessary information
- 3. Production of possible solutions
- 4. Evaluation of such solutions
- 5. Selection of a strategy for performance
- Suppose we have an input vector **x** together with a corresponding vector **t** of target variables, and our goal is to predict **t** given a new value for **x**.
- From probability perspective, we are talking about the joint distribution p(x,t).

- Determination of $p(\mathbf{x}, \mathbf{t})$ from a set of training data is *inference*.
- Taking a specific action based on the predicted/expected values of t form *Decision theory*.

Information theory

- The theory studying information gathered from known values of random variables is information theory.
- If X is a random variable with pmf p(x), then information is the quantity h(x) which is defined by Shannon as
- h(x) =-log2p(x).
- Low probability events *x* correspond to high information content.

- suppose that a sender wishes to transmit the value of a random variable to a receiver. The average amount of information that they transmit in the process is obtained by taking the expectation of with respect to the distribution $p(x)$ and is given by $H(x) = -\sum_x p(x) * \log_2 p(x)$
- called the Entropy of the random variable X.
- Consider a random variable *x* having 8 possible states, each of which is equally likely. In order to communicate the value of *x* to a receiver, we would need to transmit a message of length 3 bits. Notice that the entropy of this variable is given by
- H(x)=-8*[1/8]*log2[1/8]= 3 bits.

Probability
Basic Concepts

Random Experiment: An experiment is said to be a random experiment, if it's out-come can't be predicted with certainty.
Example; If a coin is tossed, we can't say, whether head or tail will appear. So it is a random experiment.
Sample Space: The set of all possible out-comes of an experiment is called the sample space. It is denoted by 'S' and its number of elements are n(s).
Example; In throwing a dice, the number that appears at top is any one of 1,2,3,4,5,6. So here:S ={1,2,3,4,5,6} and n(s) = 6
Similarly in the case of a coin, S={Head,Tail} or {H,T} and n(s)=2.
The elements of the sample space are called sample points or event points.
Event: Every subset of a sample space is an event. It is denoted by 'E'.
Example: In throwing a dice S={1,2,3,4,5,6}, the appearance of an event number will be the event E={2,4,6}.
Clearly E is a sub set of S.
Simple event; An event, consisting of a single sample point is called a simple event.
Example: In throwing a dice, S={1,2,3,4,5,6}, so each of {1},{2},{3},{4},{5} and {6} are simple events.
Compound event: A subset of the sample space, which has more than on element is called a mixed event.

Example: In throwing a dice, the event of appearing of odd numbers is a compound event, because E={1,3,5} which has '3' elements.
Equally likely events: Events are said to be equally likely, if we have no reason to believe that one is more likely to occur than the other.

Example: When a dice is thrown, all the six faces {1,2,3,4,5,6} are equally likely to come up.
Exhaustive events: When every possible out come of an experiment is considered.

Approaches of Probability

- Classical approach
- Frequency approach
- Subjective approach
- Axiomatic approach
  - P(A)>=0;
  - P(S)=1;
  - P(A+B)<=P(A)+P(B)

Classical definition of probability*:*
If 'S' be the sample space, then the probability of occurrence of an event 'E' is defined as:
P(E) = n(E)/N(S) = number of elements in 'E'
number of elements in sample space 'S'
Example: Find the probability of getting a tail in tossing of a coin.
Solution:
Sample space S = {H,T} and n(s) = 2
Event 'E' = {T} and n(E) = 1
therefore P(E) = n(E)/n(S) = ½
Note: This definition is not true, if
(a) The events are not equally likely.
(b) The possible outcomes are infinite.

Sure event: Let 'S' be a sample space. If E is a subset of or equal to S then E is called a sure event.
Example: In a throw of a dice, S={1,2,3,4,5,6}
Let E_1=Event of getting a number less than '7'.
So 'E_1' is a sure event.So, we can say, in a sure event n(E) = n(S)

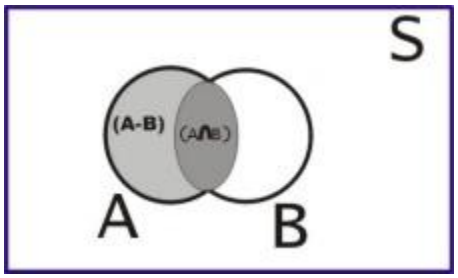
Mutually exclusive or disjoint event: If two or more events can't occur simultaneously, that is no two of them can occur together.

Addition Theorem of Probability :

If ‘A’ and ‘B’ by any two events, then the probability of occurrence of at least one of the events ‘A’ and ‘B’ is given by:

P(A or B) = P(A) + P(B) – P (A and B)

P(A∪B) = P(A) + P(B) – P (A∩B)



Problems based on addition theorem of probability:

Working rule :

(i) A ∪B denotes the event of occurrence of at least one of the event ‘A’ or ‘B’

(ii) A ∩B denotes the event of occurrence of both the events ‘A’ and ‘B’.

(iii) P(A∪B) or P(A+B) denotes the probability of occurrence of at least one of the event ‘A’ or ‘B’.

(iv) P(∩B) or P(AB) denotes the probability of occurrence of both the event ‘A’ and ‘B’.

----------x----------x----------x----------x----------x-----

Ex.: The probability that a contractor will get a contract is ‘2/3’ and the probability that he will get on other contract is 5/9 . If the probability of getting at least one contract is 4/5, what is the probability that he will get both the contracts ?

Sol.: Here P(A) = 2/3, P(B) = 5/9

P(A∪b) = 4/5, (P(A∩B) = ?

By addition theorem of Probability:

P(A∪B) = P(A) + P(B) - P(A∩B)= 4/5 = 2/3 + 5/9 - P(A∩B)

or 4/5 = 11/9 – P(A∩B)

or P(A∩B) = 11/9 – 4/5 = (55-36) / 45

P(A∩B) = 19/45

Multiplication theorem:

Let A and B be two independent events. Then multiplication theorem states that,

P[AB]= P[A]. P[B].

Note: P[AB] can also be represented by P[A and B] or P[A∩B].

Example:

Let a problem in statistics be given to two students whose probability of solving it are 1/5 and 5/7.

What is the probability that both solve the problem.

Solution:

Let A= event that the first person solves the problem.

B= event that the second person solves the problem.

It is given that P[A]=1/ 5; P[B]=5/7.

Since A and B are independent, using multiplication theorem

P[AB]= P[A]. P[B]. = 1/5*5/7= 1/7.

Conditional probability:

Probability of dependent events is termed conditional probability. Let A and B be 2 events, A depending on B. Then,

$$P[A/B] = \frac{P[A \cap B]}{P[B]}$$

Example:

Let a file contain 10 papers numbered 1 to 10. A paper is selected at random. What is the probability that it is 10 given that it is at least 5.

Solution:

From the problem we can see that,

Sample space ={1,2,3,4,5,6,7,8,9,10}

A- Event that number is 10 ={10}.

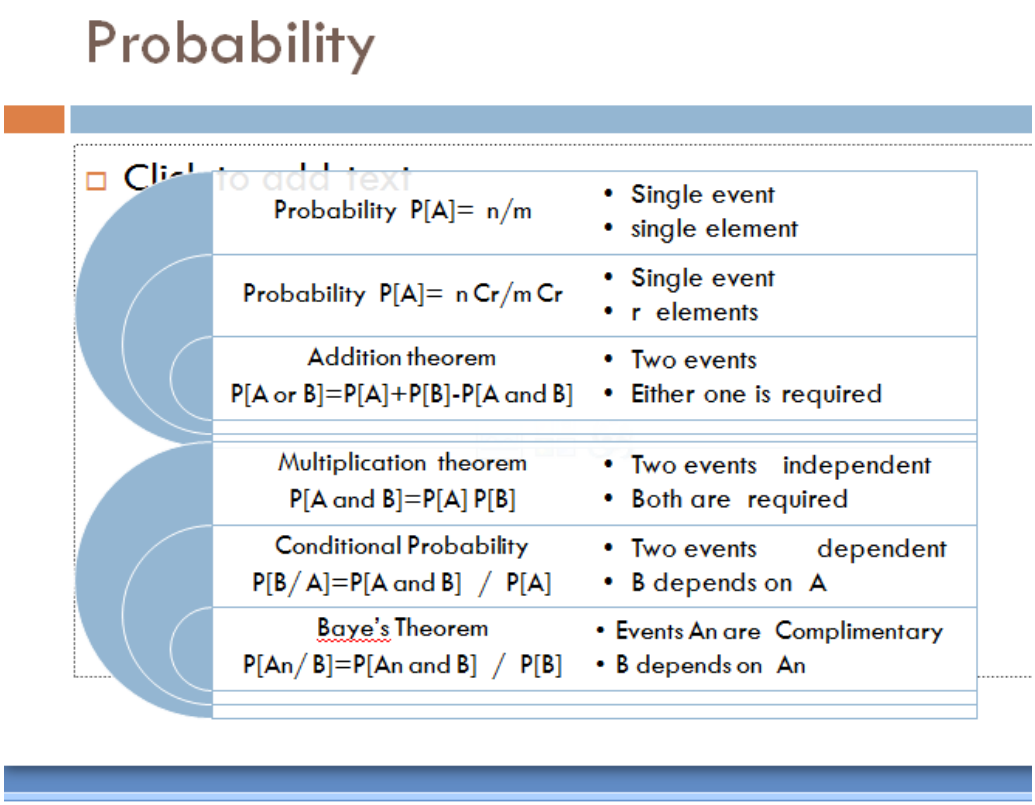
B- Event that number is at least 5 ={5,6,7,8,9,10}.

A∩B={10}.

P[A]= 1/10; P[B] =6/10; P[A∩B] =1/10.

Therefore,

$$P[A/B] = \frac{P[A \cap B]}{P[B]} = \frac{1/10}{6/10} = \frac{1}{6}$$



4. Random Variable

A random variable is a function that maps the set of events to R^n. By convention random variables are written as upper case Roman letters from the end of the alphabet like X.

For example, define the random variable X to be the sum of the two dice. For every element in the sample space, we can specify the value of X.

S={

(1; 1) = 2 (1; 2) = 3 (1; 3) = 4 (1; 4) = 5 (1; 5) = 6 (1; 6) = 7
(2; 1) = 3 (2; 2) = 4 (2; 3) = 5 (2; 4) = 6 (2; 5) = 7 (2; 6) = 8
(3; 1) = 4 (3; 2) = 5 (3; 3) = 6 (3; 4) = 7 (3; 5) = 8 (3; 6) = 9
(4; 1) = 5 (4; 2) = 6 (4; 3) = 7 (4; 4) = 8 (4; 5) = 9 (4; 6) = 10
(5; 1) = 6 (5; 2) = 7 (5; 3) = 8 (5; 4) = 9 (5; 5) = 10 (5; 6) = 11
(6; 1) = 7 (6; 2) = 8 (6; 3) = 9 (6; 4) = 10 (6; 5) = 11 (6; 6) = 12

}

If we know the probabilities of a set of events, we can calculate the probabilities that a random variable defined on those set of events takes on certain values. For example

P(X = 2) = P((1; 1)) = 1/36
P(X = 5) = P((1; 4); (2; 3); (3; 2); (4; 1g) = 1/9.
P(X = 7) = P((1; 6); (2; 5); (3; 4); (4; 3); (5; 2); (6; 1)) = 1/6
P(X = 12) = P((6; 6)) = 1/36.

The expression for P(X = 5) should be familiar, since we calculated it above as the probability of the event that the two dice sum to five. Much of the theory of probability is concerned with defining functions of random variables
and calculating the likelihood with which they take on their values.

So now we know something about what a random variable is. Now we see it a bit more closely. Random variables can be broadly classified into two types,

.Discrete r.v ---- these take only integer values

.continuous r.v ---- these can take any value

Expectation Value

Once we know the probability distribution of a random variable we can use it to predict the average outcome of functions of that variable. This is done using expectation values. The expectation value of a random variable X is defined to be

$E[(X)] = \sum_x xp(x)$ if X is discrete

$= \int xf(x)dx$ if X is continuous

The X defined in the previous section has the following mean value

E[X] = 2P(X = 2) + 3P(X = 3) + 4P(X = 4) _ _ _ + 12P(X = 12)

= 7

You can think of expectation values as taking a weighted average of the values of X where more likely values get a higher weight than less likely values.

Note: If X is continuous we do the same process where we replace $\sum$ by $\int$.

7. Variance

Once we know the probability distribution of a random variable we can use it to predict the variance of that variable. This is done using expectation values, as

$V(X) = E[X^2]-\{E[X]\}^2$

where

$E[X] = \sum_x xp(x)$ if X is discrete

$= \int xf(x)dx$ if X is continuous

$E[X^2] = \sum_x x^2p(x)$ if X is discrete

$= \int x^2f(x)dx$ if X is continuous

1**. <u>BERNOULLI:</u>**

Bernoulli trials are trials with 2 outcomes, success and failure, with

- ☐ A coin is tossed

- ☐ A die is tossed
- ☐ We write an examination

Probabilities p and q=1-p respectively. Its probability mass function is given by,

$$p(x) = \begin{cases} p & x = 1 \\ q & x = 0 \end{cases}$$

E(X) = p

VAR(X) = p(1-p)=pq .

2. **BINOMIAL:**

The random variable X denoting the number of successes in a fixed Number of independent Bernoulli trials is called a binomial random variable and its distribution is Binomial distribution as defined below

$$p(x) = nCr\ p^r q^{n-r}$$

E(X)=np

VAR(X)=np(1-p)=npq.

Example: A bag contains 50 balls of which 35 are of red colour and15 are black. 5times a ball is randomly selected , colour is noted and replaced. Find the probability that 2 times black balls are selected.

Solution:

Here every time we draw a ball it is a Bernoulli trial, as we have only 2 possibilities.

So, n=5; p=15/50; q=1-p=35/50; x=2.

$P(X=2)=\{nCxp^xq^{n-x}$

$= 5C2 \quad 15/50^2 \quad 35/50^{5-2}$

GEOMETRIC:

The random variable X denoting the number of Bernoulli trials required to achieve the first success is called a geometric random variable and its distribution is geometric distribution.

$$P(X=x) = \begin{cases} pq^{x-1} & x = 1, 2, 3 \ldots \ldots . \\ 0 \end{cases}$$

Example: A bag contains 50 balls of which 35 are of red colour and15 are black. A ball is randomly selected, if it is red it is replaced and again we select and continue till we get a black for the first time. Find the probability that we need to select 7 times before black balls is obtained.

Solution:

Here every time we draw a ball it is a Bernoulli trial, as we have only 2 possibilities.

So, x=7; p=15/50; q=1-p=35/50;

$P[X=7]=\frac{15}{50}\ \frac{35}{50}^{7-1}$

4**. POISSON:**

The random variable X whose pmf is,

$$P(X=x) = \begin{cases} (e^{-\lambda}\ \lambda^x) / x! & x = 1, 2, 3 \ldots \ldots . \\ 0 \end{cases}$$

E(X) = VAR(X) = λ.

Example: A bag contains 50 balls of which 35 are of red colour and15 are black. 20 times a ball is randomly selected , colour is noted and replaced. Find the probability that 2 times black balls are selected.
Solution:
Here every time we draw a ball it is a Bernoulli trial, as we have only 2 possibilities.
So, n=20; p=15/50; λ=np=6; x=2.

$$P(X=x) = \begin{cases} \frac{(e^{-\lambda}\lambda^x)}{x!} \end{cases} = \frac{(e^{-6}6^2)}{2!}$$

DISCUSS ABOUT CONTINUOUS DISTRIBUTIONS:

(a) **UNIFORM:**

A random variable X is uniformly distributed on the interval (a, b) if its pdf is given by,

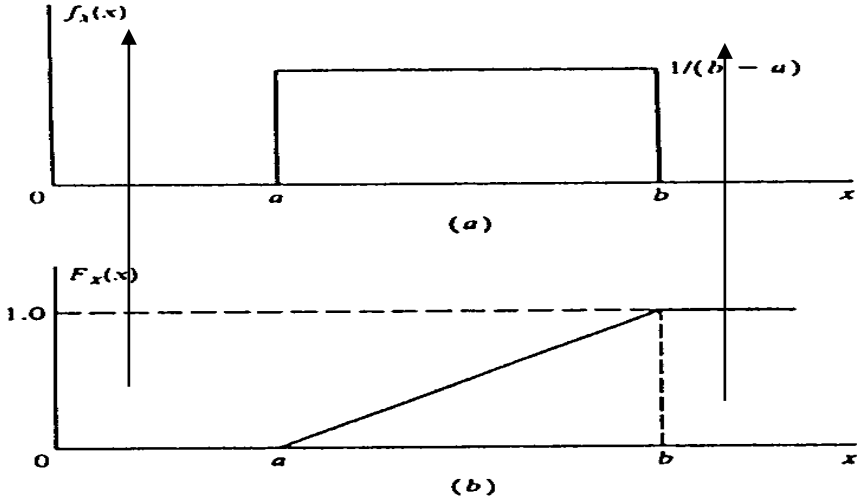
$$f(x) = \begin{cases} \frac{1}{b-a} & a \le x \le b \\ 0 & else \end{cases}$$

Its cdf is,

$$F(x) = \begin{cases} 0 & x < a \\ \frac{(x-a)}{(b-a)} & a \le x \le b \\ 1 & x > b \end{cases} \quad \{0$$

$E(X) = (a+b)/2$

$V(X)=(b-a)^2/12$



Example:

If a wheel is spun and then allowed to come to rest, the point on the circumference of the wheel that is located opposite a certain fixed marker could be considered the value of a random variable X that is uniformly distributed over the circumference of the wheel. One could then compute the probability that X will fall in any given arc.
If we assume that it is uniform in the interval[3,6], we can obtain,

Average point of outcome, $E[X]= [a+b]/12 = [3+6]/12=9/12=3/4$.
Variance $var[X]= [b-a]^2/12= [6-3]^2/12=6/12=1/2$.

2. **EXPONENTIAL:**

A Random variable X is said to be exponentially distributed if its pdf is given by,

$f(x) = \{ \lambda e^{-\lambda x}$ $x \geq 0$

$\{ 0$ otherwise

Where λ – parameter.

$f(x) = \{ 0$ $x < 0$

$\{ 1- \lambda e^{-\lambda x}$ $x \geq 0$

$E(X) = 1/\lambda$.

$V(X) = 1/\lambda^2$.

Exponential distribution is useful in representing lifetime of items, model interarrival times when arrivals are completely random and service times which are highly variable.

Exponential distribution has a property called memory less property given by,

$P(X > s + t / X > s) = p(X > t)$

This is why we are able to use exponential to model lifetimes.

Example:

Let us assume that a company is manufacturing burettes whose lifetime is assumed to be exponential with average life, 950 days. What is the probability that it is in working condition for up to 1000 days.

Solution:

It is given that , X= Lifetime of the burette , is exponential with average life 950 days i.e λ=950.

P[life time is up to 1000 days] = $P[0<X<1000] = \int_0^{1000} \lambda e^{-\lambda x}$

$= \int_0^{1000} 950\, e^{-950x}$

$= 950\, [e^{-950x}/-950]_0^{1000}$.

3. NORMAL:

A normal variable X with mean μ($-\infty < \mu < \infty$) and variance $\sigma^2 > 0$ has a normal distribution if its pdf is,

$f(x) = (1/\sqrt{2\pi}) \exp [-1/2 (x-\mu/\sigma)^2]$ $-\infty < x < \infty$

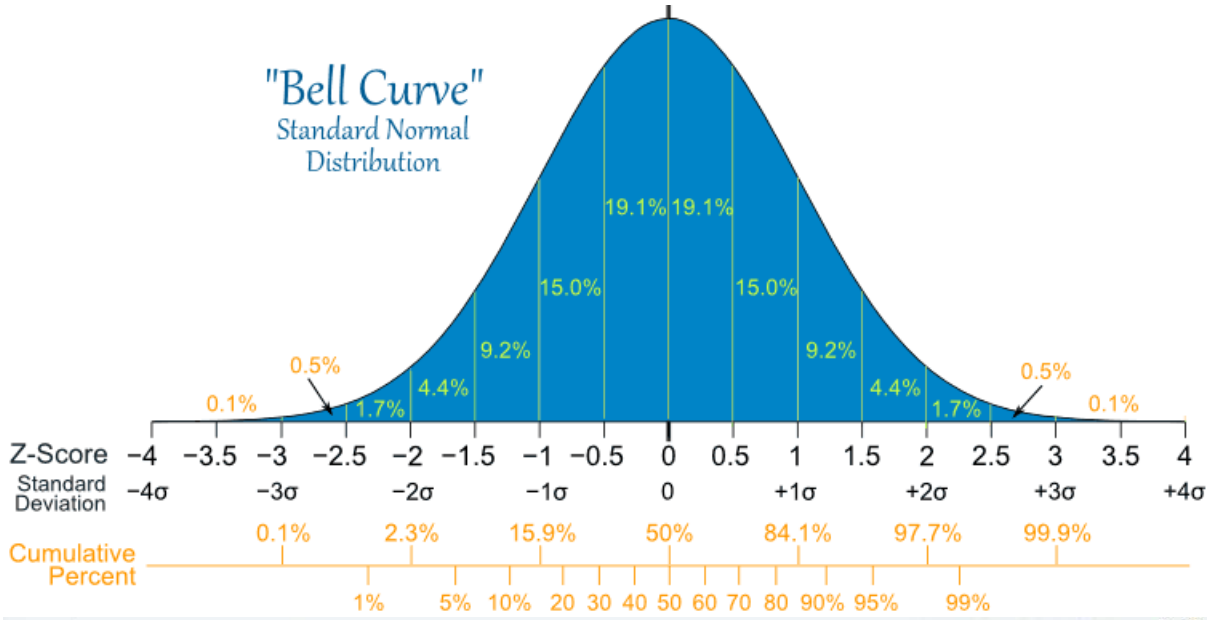
A normal distribution is used when we are having a sum of many random variables. A normal random variable with $\mu = 0$ and $\sigma = 1$ is called a standard normal r.v. Its curve is symmetrically distributed about the average $\mu = 0$.

We Standardize a normal distribution by

- Z=[X-μ]/sigma

$$p_Z(z) = (2\pi)^{-1/2} e^{-z^2/2} \qquad -\infty < z < \infty$$

- Which will give us the pdf



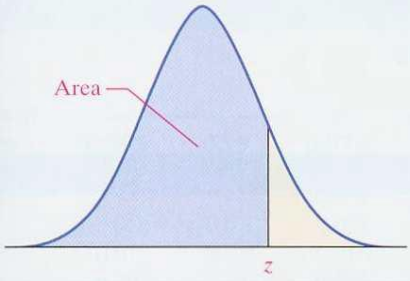


TABLE II *(continued)*

Standard Normal Distribution

z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133

Example:

Let us assume that heights of students in II M.Pharm is normally distributed with an average of 165 cm and a standard deviation of 10 cms. What is the probability that a student's height is less than 175 cms.

Solution:

Let, X= Height of students in II M.Pharm.

It is normal with, mean $\mu = 165$; standard deviation $\sigma = 10$.

P[a student's height is less than 175 cms]=P[-∞<X<175]

First, we should convert X into Z by

$Z = x - \mu / \sigma$.

We have x=175, μ= 165; σ =10.

$Z = 175 - 165 / 10 = 1$.

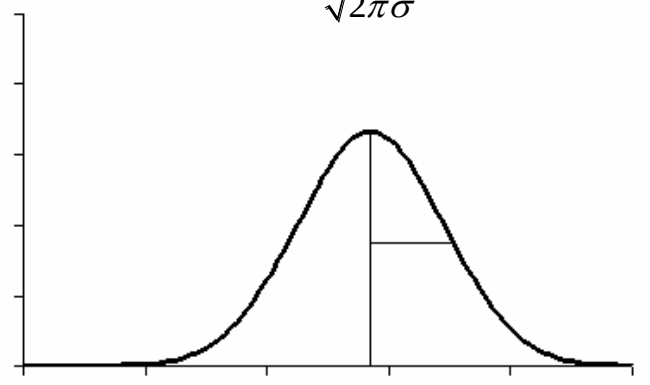
So when X=175; Z=1 and so

P[-∞<X<175] = P[-∞<Z<1]= P[-∞<Z<0]+ P[0<Z<1].

The Normal distribution

(mean μ, standard deviation σ)

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



=0.5+0.34 = 0.84.

Note:

1. The same question may have the following variations:

   P[a student's height is more than 175 cms]=P[175<X<-∞]

   = P[0<X<-∞]- P[0<X<175] =0.5- table value

   P[a student's height is between 165 and 175 cms]=P[165 < X <175]

   =P[0 < X <175]- P[0< X <165]=table value for 175 – table value for 165

Bayesian Decision Theory:

Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification

Decision making when all the probabilistic information is known

For given probabilities the decision is optimal

When new information is added, it is assimilated in optimal fashion for improvement of decisions

Example:

Suppose we have a conveyor belt carrying fish of two types Sea bass and salmon and we need the machine to identify them and pack.

Soln:

Let w=State of nature, so that w1= sea bass and w2=salmon.

a Priori probability P(w1): probability that next fish in line is sea bass;

P(w2): probability that next fish in line is salmon;

We need features for classification like Length, Lightness, width, Number and shape of fins, Position of the mouth. If we have character x, then P(wi/x) is the conditional probability after measuring the feature, also called 'a Posteriori'. There is one more conditional probability here, which is P(x/wi) which is called the 'Likelihood' and the probability(x) , called the 'Evidence', makes up therequired set for the Bayesian inference

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}}$$

The procedure is very simple. Calculate the 'a Posteriori' probabilities and classify the element into the class with the high 'a Posteriori' w1 if P(w1/x)>=P(w2/x), else w2.

Naïve Baye's:

- Prior, conditional and joint probability for random variables
  - Prior probability: P(x)
  - Conditional probability: $P(x_1 \mid x_2), P(x_2 \mid x_1)$
  - Joint probability: $\mathbf{x} = (x_1, x_2), P(\mathbf{x}) = P(x_1, x_2)$
  - Relationship: $P(x_1, x_2) = P(x_2 \mid x_1)P(x_1) = P(x_1 \mid x_2)P(x_2)$
  - Independence: $P(x_2 \mid x_1) = P(x_2), P(x_1 \mid x_2) = P(x_1), P(x_1, x_2) = P(x_1)P(x_2)$
- Bayesian Rule $P(c \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c)P(c)}{P(\mathbf{x})}$ $\quad Posterior = \frac{Likelihood \times Prior}{Evidence}$

Naive Bayes Examples

We are going to use the Iris dataset we used in the previous blogs to illustrate how Naive Bayes works. Let's suppose we measure an Iris Setosa and find the following measurements:

- Sepal length = 7 cm
- Sepal width = 3 cm
- Petal length = 5 cm
- Petal width = 2 cm

From our data we know that each class—Iris versicolor, Iris virginica, and Iris setosa—represents one-third of the data. Following Naive Bayes, we need to calculate the following conditional probabilities and categorize our measured flower with the class that has the highest probability. To do so, we are going to use the Gaussian measure of likelihood:

- P(Setosa | 7,3,5,2)
- P(Versicolor | 7,3,5,2)
- P(Virginica | 7,3,5,2)

Let's do one calculation with Versicolor:

- P(Versicolor | 7,3,5,2) = (P(7,3,5,2 | Versicolor) * P(Versicolor))/ P(7,3,5,2)

Assuming independence and using the Gaussian distribution of conditional class probabilities, we can calculate the following:

- P(7,3,5,2 | Versicolor) = P(7 | Versicolor) * P(3 | Versicolor) * P(5 | Versicolor) * P(2 | Versicolor)

To calculate each of the conditional class probabilities, we have to find the average and standard deviation of each of the features operating under the assumption that they are Versicolor. The average and standard deviation are calculated as follows.

	Sepal Length	Sepal Width	Petal Length
Average	5.936	2.77	4.26
Standard Deviation	0.51	0.31	0.46

Plugging those values into a Gaussian distribution, we can calculate: (N stands for the normal distribution)

P(7,3,5,2 | Versicolor) = P(7 | Versicolor) * P(3 | Versicolor) * P(5 | Versicolor) * P(2 | Versicolor)

P(7,3,5,2 | Versicolor) = N(7 | 5.936, 0.51) * N(3 | 2.77, 0.31) * N(5 | 4.26, 0.46) * N(2 | 1.32, 0.19)

P(7,3,5,2 | Versicolor) = 0.089 * 0.97 * 0.24 * 0.05

P(7,3,5,2 | Versicolor) = 0.001

P(Versicolor) = 50/150

At last we can calculate: P(7,3,5,2 | Versicolor) 0.001* 0.33 = .0003

From here, we would need to calculate the same process for Setosa and Virginica in order to determine in which class the original flower is most likely to belong.

BAYESIAN CLASSIFIER

- D : Set of tuples
  - Each Tuple is an 'n' dimensional attribute vector
  - X : $(x_1,x_2,x_3,\ldots. x_n)$
  - where x_i is the value of attribute A_i
- Let there are 'm' Classes : $C_1,C_2,C_3 \ldots C_m$
- Bayesian classifier predicts X belongs to Class C_i iff
  - $P(C_i|X) > P(C_j|X)$ for $1 <= j <= m$, $j \neq i$
- Maximum Posteriori Hypothesis
  - $P(C_i|X) = \dfrac{P(X|C_i)\,P(C_i)}{P(X)}$
  - Maximize $P(X|C_i)\,P(C_i)$ as $P(X)$ is constant

MLE-MAP:

- The learner considers some set of candidate hypotheses H and it is interested in finding the ***most probable hypothesis*** h **in** H given the observed data D.
- Any such maximally probable hypothesis is called a ***maximum a posteriori (MAP) hypothesis*** h_{MAP}.

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}} P(h|D)$$

$$= \underset{h \in H}{\operatorname{argmax}} \frac{P(D|h)P(h)}{P(D)}$$

$$= \underset{h \in H}{\operatorname{argmax}} P(D|h)P(h)$$

Example:

- Hypothesis space H
- Hypothesis space H; H={ h1, h2, h3};
- h1->multiples of 10;h2->even numbers;h3->odd numbers
- Likelihood p(D|h)
- Let X={20,40,60}
- X={20,40,60}
- H1 = multiples of 10 = {10,20,...,100}
- H2 = even numbers = {2,4,...,100}
- H3 = odd numbers = {1,3,...,99}
- P(X|H1) = 1/10 * 1/10 * 1/10=1/1000
- p(X|H2) = 1/50 * 1/50 * 1/50=1/125000
- P(X|H3) = 0

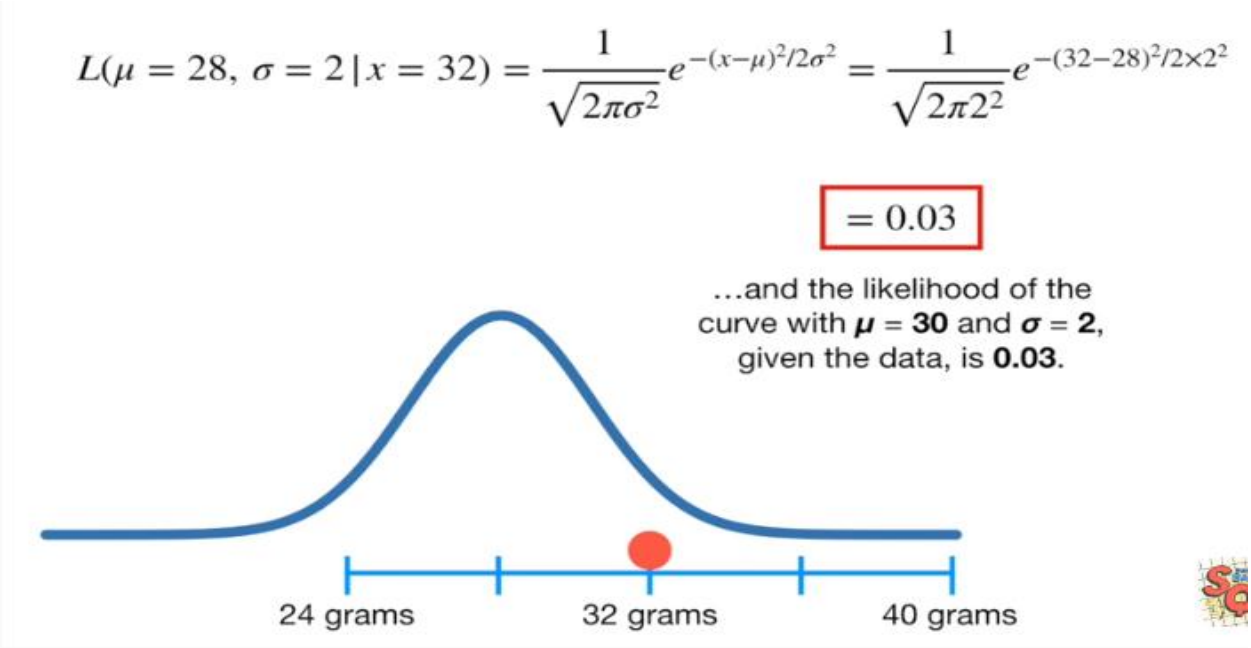
- Hypothesis space H;
- H={ h1, h2, h3}
- h1->multiples of 10;h2->even numbers;h3->odd numbers
- Prior p(h)– Assume all hypotheses are equally likely
- Likelihood p(D|h)
- Algorithm for computing posterior p(h|D)
- p(h|D)={ p(D|h)*p(h)} /p(D)
- p(h1|Data=X)={1/1000*1/3}/p(X)=[1/3000]/0.000336
- =0.99107
- Similarly p(h2/X) =0.0079 and p(h3/X)=0
- Select hi with higher probability , giving h1 as the hypothesis

Event	Prior	Posterior
h1	0.33	0.99107

h2	0.33	0.0079
h3	0.33	0

- Difference between probability and Likelihood.
- Assume we are in Gaussian domain. That is , say, heights of students in this class is normally distributed with mean 160 and SD=25.
- Then we say probability of a randomly selected student's height to be 165 is (from area ideas)=0.20.
- On the other hand , if we do not know the exact parameters of the distribution and we have a student's height is known as 165. Then we'll ask, what is the Likelihood that it is N(160,25)?

- Suppose we have x=32. If we assume mean=28 and SD=2, then , the above equation gives L=0.03



- In the above discussion we had a single sample x=32. If there are two samples assuming indpt, then the Likelihood of the parameters are
- L[mean=30,SD=2|x1=32, x2=35]=
- L[mean=30,SD=2|x1=32]* L[mean=30,SD=2| x2=35]
- Generalizing, we get

$$L(\mu, \sigma | x_1, x_2, \ldots, x_n) = L(\mu, \sigma | x_1) \times \ldots \times L(\mu, \sigma | x_n)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_1-\mu)^2/2\sigma^2} \times \ldots \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_n-\mu)^2/2\sigma^2}$$

$$h_{ML} = \underset{h \in H}{\text{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\alpha^2}(d_i-\mu)^2}$$

$$= \underset{h \in H}{\text{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\alpha^2}(d_i-h(x_i))^2}$$

$$= \underset{h \in H}{\text{argmax}} \sum_{i=1}^{m} ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\alpha^2}(d_i - h(x_i))^2$$

$$= \underset{h \in H}{\text{argmax}} \sum_{i=1}^{m} -\frac{1}{2\alpha^2}(d_i - h(x_i))^2$$

$$= \underset{h \in H}{\text{argmin}} \sum_{i=1}^{m} \frac{1}{2\alpha^2}(d_i - h(x_i))^2$$

$$= \underset{h \in H}{\text{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

- Since the function on RHS is in product form with many terms, to simplify we take logarithm and fine minimum using first derivative method.

Example:

- Consider a sample 0,1,0,0,1,0 from a binomial distribution, with the form P[X=0]=(1-p), P[X=1]=p. Find the maximum likelihood estimate of p.
- Soln :
- L(p)=P[X=0] P[X=1] P[X=0] P[X=0] P[X=1] P[X=0]
- =(1-p) p (1-p) (1-p) p (1-p)
- $=(1-p)^3p^2$.
- Log L(p)=log[$(1-p)^3p^2$.]=3log(1-p)+2logp
- Log L(p)=log[$(1-p)^3p^2$.]=3log(1-p)+2logp.
- To find minimum, find derivative w.r.t p, equate it to zero to get p=2/5.

- Is this maximum or minimum?
- To find it, get the second derivative , substitute to find out.

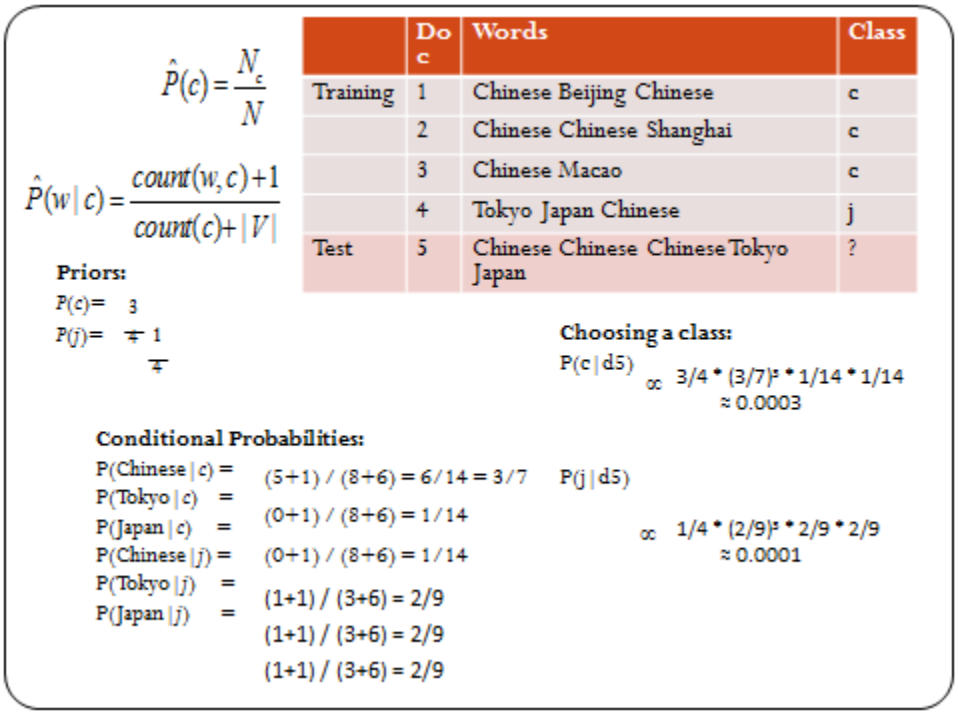
Naïve Baye's Classifier-Text classification
Attributes are text positions, values are words

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}}\ P(c_j)\prod_i P(x_i \mid c_j)$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\ P(c_j)P(x_1 = \text{"our"} \mid c_j)\cdots P(x_n = \text{"text"} \mid c_j)$$

- From training corpus, extract *Vocabulary*
- Calculate required $P(c_j)$ and $P(x_k \mid c_j)$ terms
  - For each c_j in C do
    - $docs_j \leftarrow$ subset of documents for which the target class is c_j
  - $P(c_j) \leftarrow \dfrac{|docs_j|}{|\text{total \# documents}|}$
    - $Text_j \leftarrow$ single document containing all $docs_j$
    - for each word x_k in *Vocabulary*
      - $n_k \leftarrow$ number of occurrences of x_k in $Text_j$

$$P(x_k \mid c_j) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$$



$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = 3/4$

$P(j) = 1/4$

Choosing a class:

$P(c \mid d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14 \approx 0.0003$

Conditional Probabilities:

$P(\text{Chinese} \mid c) = (5+1)/(8+6) = 6/14 = 3/7$

$P(\text{Tokyo} \mid c) = (0+1)/(8+6) = 1/14$

$P(\text{Japan} \mid c) = (0+1)/(8+6) = 1/14$

$P(\text{Chinese} \mid j) = (1+1)/(3+6) = 2/9$

$P(\text{Tokyo} \mid j) = (1+1)/(3+6) = 2/9$

$P(\text{Japan} \mid j) = (1+1)/(3+6) = 2/9$

$P(j \mid d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9 \approx 0.0001$

Bayes Optimal classifier:

- MAP ,ML, Bayes.. discuss the most probable or representative for the data. But that is to get the model ready. Its purpose , like classification, needs the models efficiency on any new instance and the most probable classification need not be the most likely one or MAP.
- consider a hypothesis space containing three hypotheses, hl, h2, and h3 with posteriors 0.4,0.3 and 0.3 resply. The MAP is h1. Suppose a new instance x1 is classified positive by h1 and negative by h2 and h3, then the most likely classification is negative (h2+h3)

- Hence, the most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities.
- If the possible classification of the new example can take on any value ***vj*** from some set V, then the probability P(vjlD) that the correct classification for the new instance is vj, is

$$P(v_j \mid D) = \sum_{h_i \in H} P(v_j \mid h_j) P(h_j \mid D)$$

- The optimal classification of the new instance is the value ***vj,*** for which P (vj | D) is maximum.

$$Bayes\ opt\ class = \arg\max_{v_j \in V} P(v_j \mid D) = \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_j) P(h_j \mid D)$$

Gibbs Algorithm:

- 1. Choose a hypothesis ***h*** from H at random, according to the posterior probability distribution over H.
- 2. Use ***h*** to predict the classification of the next instance ***x.***

Minimum Description length Principle:
From MAP discussion, we know that

$$\begin{aligned} h_{MAP} &\equiv \underset{h \in H}{\operatorname{argmax}}\ P(h|D) \\ &= \underset{h \in H}{\operatorname{argmax}}\ \frac{P(D|h)\,P(h)}{P(D)} \\ &= \underset{h \in H}{\operatorname{argmax}}\ P(D|h)\,P(h) \end{aligned}$$

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\ \log_2 P(D|h) + \log_2 P(h)$$

$$h_{MAP} = \underset{h \in H}{\operatorname{argmin}}\ -\log_2 P(D|h) - \log_2 P(h)$$

- - log, ***P(h)*** is the description length of ***h*** under the optimal encoding for the hypothesis space H.
- -log2 ***P(D1h)*** is the description length of the training data ***D*** given hypothesis ***h,*** under its optimal encoding.

- Shannon –
- the optimal code (i.e., the code that minimizes the expected message length) assigns - log, ***pi*** bitst to encode message ***i .***
- minimize the expected code length we should assign shorter codes to messages that are more probable.
- number of bits required to encode message ***i*** using code ***C*** as the ***description length of message i with respect to C,*** which we denote by ***Lc(i)***

PROBABILITY – PRACTICE PROBLEMS

1. The probability that a contractor will get a contract is '2/3' and the probability that he will get on other contract is 5/9 . If the probability of getting at least one contract is 4/5, what is the probability that he will get both the contracts ?: soln 19/45.
2. Let a problem in statistics be given to two students whose probability of solving it are 1/5 and 5/7. What is the probability that both solve the problem. Solution: 1/7
3. A coin is thrown 3 times .what is the probability that atleast one head is obtained? Soln:7/8.
4. What is the probability of getting a sum of 7 when two dice are thrown?. Soln:1/6.
5. 1 card is drawn at random from the pack of 52 cards.
   (i) Find the Probability that it is an honor card. Soln:16/52
   (ii) It is a face card. Soln: 12/52
6. Three dice are rolled together. What is the probability as getting at least one '4'?.soln:91/216.
7. A problem is given to three persons P, Q, R whose respective chances of solving it are 2/7, 4/7, 4/9 respectively. What is the probability that the problem is solved? Soln:122/147.
8. Find the probability of getting two heads when five coins are tossed. Soln:5/16.
9. What is the probability of getting a sum of 22 or more when four dice are thrown?.soln:15/1296.
10. Two dice are thrown together. What is the probability that the number obtained on one of the dice is multiple of number obtained on the other dice?.soln: 11/18.
11. From a pack of cards, three cards are drawn at random. Find the probability that each card is from different suit. Soln:[$4*13^3/52C3$].
12. Find the probability that a leap year has 52 Sundays. Soln: 5/7.
13. Three bags contain 3 red, 7 black; 8 red, 2 black, and 4 red & 6 black balls respectively. 1 of the bags is selected at random and a ball is drawn from it. If the ball drawn is red, find the probability that it is drawn from the third bag. Soln: 4/15
14. A coin is tossed 4 times. What is the probability of getting atleast 2 heads.
15. Marie is getting married tomorrow, at an outdoor ceremony in the desert. In recent years, it has rained only 5 days each year. Unfortunately, the weatherman has predicted rain for tomorrow. When it actually rains, the weatherman correctly forecasts rain 90% of the time. When it doesn't rain, he incorrectly forecasts rain 10% of the time. What is the probability that it will rain on the day of Marie's wedding?
16. Information is transmitted digitally as binary sequences, the bits. However, noise on the channel corrupts the signal, in that a digit is interchanged with a probability (1-p) , for both the digits. Based on transmission records, the 0's and 1's are transmitted in the ratio 3:4. Find the probability that at a particular time, we receive the signal '10'.
17. A company produces a product with 4 machines with defective probabilities .2,.3,.4,.5. If a randomly inspected product is defective, what are the chances of it to have been produced by A1, knowing that all machines produce equally same number of products.
18. A company produces a product with 3 machines with defective probabilities .35,.45,.20. If a randomly inspected product is defective, what are the chances of it to have been produced by A2, knowing that all machines produce equally same number of products.

BITS WILP Machine Learning (ISZC464) Quiz-1 2017-H2

Q1.
Which of the following statements are true in context of Graphical models?
Select one or more:
a. None of the above.
b. Bayesian Belief networks describe conditional independence among subsets of variables.
c. Bayes network represents the joint probability distribution over a collection of random variable.
d. Each node denotes a random variable.
Answer: (B, C, D)

A **Bayesian network**, **Bayes network**, **belief network**, **Bayes(ian) model** or **probabilistic directed acyclic graphical model** is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).

A Bayesian belief network describes the probability distribution governing a set of variables by specifying a set of conditional independence assumptions along with a set of conditional probabilities. In contrast to the naive Bayes classifier, which assumes that *all* the variables are conditionally independent given the value of the target variable, Bayesian belief networks allow stating conditional independence assumptions that apply to *subsets* of the variables. Thus, Bayesian belief networks provide an intermediate approach that is less constraining than the global assumption of conditional independence made by the naive Bayes classifier, but more tractable than avoiding conditional independence assumptions altogether. Bayesian belief networks are an active focus of current research, and a variety of algorithms have been proposed for learning them and for using them for inference.

Q2.
Assuming log base 2, the entropy of a binary feature with p(x=1) = 0.5 is
Select one:
a. 0.75
b. 0
c. 0.25
d. 1
e. 0.5
Answer: (D) It is '1'.

Entropy

Let X be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x)$

$$p(x) = Pr\{X = x\}, \quad x \in \mathcal{X}$$

The *entropy* of the variable X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

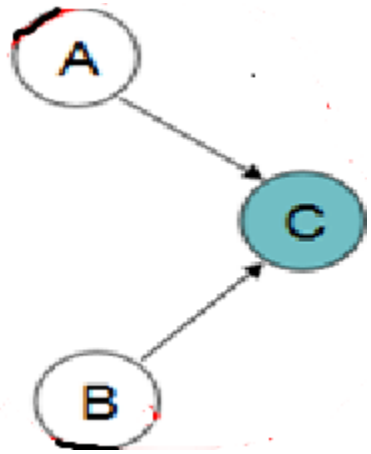
The logarithm can be in any base, but normally base 2 is used. The unit of the entropy is then called *bits*. If we use base b in the logarithm, we denote the entropy by $H_b(X)$. We can easily convert between entropies in different bases

$$H_b(X) = \log_b a \cdot H_a(X)$$

By convention $0 \log 0 = 0$, since $y \log y \to 0$ as $y \to 0$.

The entropy is a measure of the information content of a random variable.

Q3.
Which of the following statement are true for the given graphical model.



Select one or more:
a. A is conditionally independence of B given C.
b. B is conditionally independence of A given C.
c. B is not conditionally independence of A given C.
d. A is not conditionally independence of B given C.
Answer: C, D
(B is not conditionally independence of A given C., A is not conditionally independence of B given C.)

Q4.
Let X be random variable and let Y=aX+b, where a and b are given scalars. Then which of the following statements are true.
(E[Z] states the expected value of Z)
Select one or more:
a. E[Y]=(a/b)*E[X]
b. E[Y] = E[X]
c. E[Y]=a*E[X]+b
d. E[Y]=a*b*E[X]

Expected value of a constant is constant [edit]

If c is a constant random variable, then $\mathrm{E}[c] = c$. This implies that for any random variable X, $\mathrm{E}[\mathrm{E}[X]] = \mathrm{E}[X]$.

Linearity [edit]

The expected value operator (or **expectation operator**) $\mathrm{E}[\cdot]$ is linear in the sense that

$$\mathrm{E}[X + Y] = \mathrm{E}[X] + \mathrm{E}[Y],$$

$$\mathrm{E}[aX] = a\,\mathrm{E}[X],$$

where X and Y are (arbitrary) random variables, and a is a scalar.

If a and b are constants then $\mathrm{Var}(aX+b) = a^2\mathrm{Var}(X)$

$\mathrm{E}(aX+b) = a\,\mathrm{E}(X) + b$
$\mathrm{Var}(aX+b) = \mathrm{E}[(aX+b - (a\,\mathrm{E}(X)+b))^2] = \mathrm{E}(a^2(X - \mathrm{E}(X))^2) = a^2\mathrm{E}((X - \mathrm{E}(X))^2) = a^2\mathrm{Var}(X)$

The square root of Var(X) is called the standard deviation of X.
SD(X) = sqrt(Var(X)): measures scale of X.

Answer: (C)

Q5.
When we can use Expectation maximization algorithm.
Select one or more:
a. None of the these.
b. Unsupervised clustering (target value unobservable).
c. Data is only partially observable.
d. Supervised Learning (some instance attributes unobservable).
Answer: B, C, D

Expectation Maximization (EM) Algorithm

- When to use
  - ✓ Data is only partially observable
  - ✓ Unsupervised clustering (target value unobservable)
  - ✓ Supervised Learning (some instance attributes unobservable)

- Some uses
  - ✓ Train Bayesian Belief Networks
  - ✓ Unsupervised clustering
  - ✓ Learning Hidden Markov Models

Q6.
Which of the following statements are true?
Select one or more:
a. Maximum a Posteriori estimation seek the estimate of Θ that is most probable, given the observed data, plus background assumptions about its value.
b. Maximum Likelihood estimation seek the estimate of Θ that is most probable, given the observed data, plus background assumptions about its value.
c. Maximum Likelihood estimation seek an estimate of Θ that maximizes the probability of the observed data.
d. Maximum a Posteriori estimation seek an estimate of Θ that maximizes the probability of the observed data.
Answer: A, C

MAP principle: We should choose the value of θ that is most probable, given the observed data D and our prior assumptions summarized by P(θ); that is

$$\hat{\theta}^{MAP} = \arg\max_{\theta} P(\theta|D)$$

...

Maximum Likelihood Estimation (MLE)

- We should choose the value of θ that makes data set, D most probable

$$\hat{\theta}^{MLE} = \arg\max_{\theta} P(D|\theta)$$

In MLE, we don't have prior knowledge, as in the example of a toss of coin, about the coin whether it is biased or unbiased. We arrive at Theta based on the data.
While in MAP, we incorporate our prior knowledge:

Second Algorithm allow us to incorporate our **prior knowledge** about coins by adding any number of imaginary coin flips resulting in heads and tails.

Let assume, γ_1 denotes imaginary heads
γ_0 denotes imaginary tails

Considering this prior knowledge, now the $\ddot{\Theta}$ can be estimated as follows:

$$\ddot{\Theta} = \alpha_1 + \gamma_1 / (\alpha_1 + \gamma_1 + \alpha_0 + \gamma_0)$$

Q7.
If X is a vector of n attributes and Y is boolean valued label. How many different functions are possible? (2^n represents 2^n)
a. 2^{2^n}
b. 2^{n^2}
c. 2^n
d. 2n
Answer: A

It is 2^(2^n)
If X has two attributes x1 and x2, then # of observations one has to take are (x1=0, x2=0, y), (x1=0, x2=1, y), (x1=1, x2=0, y), (x1=1, x2=1, y). 'n' attributes means 2^n states.

Number of functions would be: 2^(n^2)

X1, X2, Y
Function: 1
0, 0, 0
0, 1, 0
1, 0, 0
1, 1, 0
Function: 2
0, 0, 0
0, 1, 0
1, 0, 0
1, 1, 1

Run 1:
Input: X1, X2
0, 0
0, 1
1, 0
1, 1
Output: 0,0,0,0

Run 2:
Input: X1, X2
0, 0
0, 1
1, 0
1, 1

Output: 0,0,0,1

Run 3:
Input: X1, X2
0, 0
0, 1
1, 0
1, 1
Output: 0,0,1,0

Run 4:
Input: X1, X2
0, 0
0, 1
1, 0
1, 1
Output: 0,0,1,1

Run 5: Output: 0,1,0,0. Run 6: Output: 0,1,0,1. Run 7: Output: 0,1,1,0. Run 8: Output: 0,1,1,1
Run 9: Output: 1,0,0,0. Run 10: Output: 1,0,0,1. Run 11: Output: 1,0,1,0. Run 12: Output: 1,0,1,1
Run 13: Output: 1,1,0,0. Run 14: Output: 1,1,0,1. Run 15: Output: 1,1,1,0. Run 16: Output: 1,1,1,1

This can be understood as there will be 2^n rows in one truth table for X1, X2. Now, assume Y to be a vector of length 2^n, number of states it can take = 2^(2^n).

Q8.
Which of the following statements are true?
Select one or more:
a. To infer posterior probability, Bayesian linear regression uses Naïve Bayes principle.
b. None of these
c. Bayesian linear regression cannot be used for classification.
d. In Bayesian linear regression Prior can be used for regularization.
Answer: A, D
In Bayesian linear regression Prior can be used for regularization., To infer posterior probability, Bayesian linear regression uses Naïve Bayes principle.

Bayesian Linear Regression

innovate achieve

- $D = \{(x_1,y_1), (x_2,y_2),\ldots,(x_N,y_N)\}$ where $x_i \in \mathbb{R}^D$ and $y \in \mathbb{R}$
- Model : $Y_1, Y_2, \ldots, Y_N$ independent given $\mathbf{w}$, $Y \sim \mathcal{N}(w^T x_i, \beta)$ [β is precision; $\beta = 1/\sigma^2$]
- $\mathbf{w} \sim \mathcal{N}(0, \alpha^{-1}I)$ where $\mathbf{w} = (w_0, w_1, \ldots, w_D)^T$
- Assume β and α are known
  - ✓ therefore only unknown parameter is $\mathbf{w}$
- **Likelihood:**

$$p(\boldsymbol{D}|\boldsymbol{w}) \propto exp\left(-\frac{\beta}{2}(\boldsymbol{y}-\boldsymbol{Qw})^T(\boldsymbol{y}-\boldsymbol{Qw})\right)$$

- **Posterior:**

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto p(\boldsymbol{D}|\boldsymbol{w})\, p(\boldsymbol{w})$$

...

Posterior of w

innovate achieve lead

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto p(\boldsymbol{D}|\boldsymbol{w})\, p(\boldsymbol{w})$$

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto exp\left(-\frac{\beta}{2}(\boldsymbol{y}-\boldsymbol{Qw})^T(\boldsymbol{y}-\boldsymbol{Qw})\right) exp\left(-\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}\right)$$

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto exp\left(-\frac{\beta}{2}(\boldsymbol{y}-\boldsymbol{Qw})^T(\boldsymbol{y}-\boldsymbol{Qw}) - \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}\right)$$

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto exp\left(-\frac{1}{2}(\beta(\boldsymbol{y}-\boldsymbol{Qw})^T(\boldsymbol{y}-\boldsymbol{Qw}) + \alpha\boldsymbol{w}^T\boldsymbol{w})\right)$$

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto exp\left(-\frac{1}{2}\left(\beta(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{Qw})\right) + \alpha\boldsymbol{w}^T\boldsymbol{w})\right) \quad [-2\boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{y} = -\boldsymbol{y}^T\boldsymbol{Qw} - (\boldsymbol{Qw})^T\boldsymbol{y}]$$

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto exp\left(-\frac{1}{2}\left((\beta\boldsymbol{y}^T\boldsymbol{y} - 2\beta\boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{y} + \beta\boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{Qw})\right) + \alpha\boldsymbol{w}^T\boldsymbol{w})\right)$$

$$p(\boldsymbol{w}|\boldsymbol{D}) \propto exp\left(-\frac{1}{2}((\beta\boldsymbol{y}^T\boldsymbol{y} - 2\beta\boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{y} + \boldsymbol{w}^T(\beta\boldsymbol{Q}^T\boldsymbol{Q} + \alpha\boldsymbol{I})\boldsymbol{w}\right)$$

...

Posterior of w

$$p(\boldsymbol{w}|\boldsymbol{D})\ \alpha\ exp\left(-\frac{1}{2}\left(\left(\beta \boldsymbol{y}^T\boldsymbol{y}\ -2\beta \boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{y}\ +\boldsymbol{w}^T(\beta \boldsymbol{Q}^T\boldsymbol{Q}+\alpha \boldsymbol{I})\boldsymbol{w}\right)\right) \quad :::1$$

Below we are writing a multi-variate Gaussian distribution:

Completing the square:

$$\mathcal{N}(\boldsymbol{\mu}, \Lambda^{-1})\ \alpha\ \boldsymbol{exp}(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu})^T\Lambda(\boldsymbol{w}-\boldsymbol{\mu}))$$

$$\mathcal{N}(\boldsymbol{\mu}, \Lambda^{-1})\ \alpha\ \boldsymbol{exp}(-\frac{1}{2}(\boldsymbol{w}^T\Lambda\boldsymbol{w}\ -\boldsymbol{w}^T\Lambda\boldsymbol{\mu}-\boldsymbol{\mu}^T\Lambda\boldsymbol{w}+\boldsymbol{\mu}^T\Lambda\boldsymbol{\mu}))$$

$$\mathcal{N}(\boldsymbol{\mu}, \Lambda^{-1})\ \alpha\ \boldsymbol{exp}(-\frac{1}{2}(\boldsymbol{w}^T\Lambda\boldsymbol{w}\ -2\boldsymbol{w}^T\Lambda\boldsymbol{\mu}+\boldsymbol{\mu}^T\Lambda\boldsymbol{\mu})) \quad :::2$$

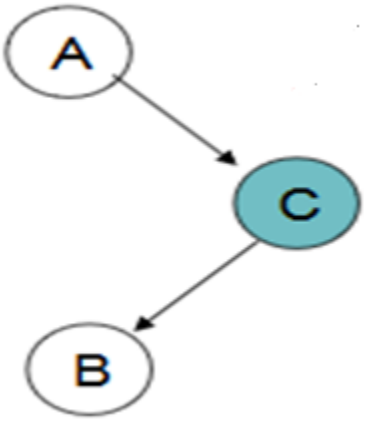
On matching expression (1) and (2), we get:

$$\Lambda = \beta \boldsymbol{Q}^T\boldsymbol{Q}+\alpha \boldsymbol{I}$$

$$\boldsymbol{\mu} = \beta\Lambda^{-1}\boldsymbol{Q}^T\boldsymbol{y} \quad \text{(Using } \boldsymbol{w}^T\Lambda\boldsymbol{\mu} = \beta \boldsymbol{w}^T\boldsymbol{Q}^T\boldsymbol{y})$$

These slides show the derivation of posterior probability using Bayes theorem, and here all probabilities are represented by multivariate Gauss distribution.

Q9.
Which of the following statement are true for the given graphical model?



Select one or more:
a. B is not conditionally independence of A given C.
b. A is conditionally independence of B given C.
c. B is conditionally independence of A given C.
d. A is not conditionally independence of B given C.
Answer: B, C

Conditional independence

- Let A, B, and C be events. A and B are *conditionally independent given* C iff

$$P(A \mid C) = P(A \mid B \cap C)$$

or, equivalently, iff

$$P(A \cap B \mid C) = P(A \mid C) \times P(B \mid C)$$

- If A and B are conditionally independent, then once we learn C, learning B gives us no *additional* information about A.
- Two random variables X and Y are conditionally independent given Z if for all x, y, and z

$$p_{X|Z}(x, z) = p_{X|YZ}(x, y, z)$$

In this case we write $X \perp\!\!\!\perp Y \mid Z$. This also corresponds to (perhaps infinitely) many event conditional independencies.

...

Head to Tail

- Prove A conditional independence of B given C?

Proved in : P(A,B | C) = P(A,B,C)/P(C) = P(A)*P(C|A)*P(B|A)/P(C) = P(A|C) * P(B|C)
Hence, P(A,B | C) = P(A|C) * P(B|C)

This implies “A and B are conditionally independent given C”.

Q10.
Smoothing can be used in which of the following cases:
Select one or more:
a. When likelihood estimates zero probability
b. When test error and training error are very different
c. None of the above
d. When learning algorithm result in very rough function
Answer: A
(When likelihood estimates zero probability.)
Could someone explain Laplacian smoothing (or 1-up smoothing)?
Ans: Suppose you are looking at outcomes of a die. Let us say you get the following outcomes of each number, in 10 throws:
One : 1
Two : 3
Three : 1
Four : 0
Five : 3
Six : 2

Now, the probabilities without the smoothing are
One : 1/10
Two : 3/10
Three : 1/10
Four : 0/10
Five : 3/10
Six : 2/10

The sums of probabilities is (of course) 1.

To smoothen out, we add '1' to numerators. Now we need to add "something" to the denominator such that the sum remains 1. So,
$(1+1+3+1+1+1+0+1+3+1+2+1) / (10+K) = 1$

This gives K=6. Now note that if you had zero throws, the probabilities are all 1/6. These are called the "prior probabilities" - our prior assumption of the outcomes. We initially believe all of them are equally likely.
And K=6 is essentially the no. of classes!
(URL: https://www.quora.com/Could-someone-explain-Laplacian-smoothing-or-1-up-smoothing)

Q11.
Which of the following statements are true in context of decision trees?
Select one or more:
a. Capable in classifying non-linearly separable data.
b. None of these.
c. Capable in classifying linearly separable data.
d. It is always possible to get zero training error.
Answer: A, C, D
"Zero training error": means decision tree can give a model that will give correct output for all the traning data.
An attribute is a discreete valued variable. While traversing a decision tree downwards based on attributes, it is always possible to arrive at a label (decision (yes, no)).

Q12.
Let a probability of disease is 1 in 10,000 and the test accuracy of the disease is 99 %. Let event A is the event you have this disease, and event B is the event that you test positive. Given test is positive what is the probability that disease is actually present? Precisely you need calculate probability P(A|B)
Select one:
a. 0.0990
b. 0.0988
c. 0.9902
d. 0.0098
Answer: (D = 0.0098)

$P(B|A) = \frac{99}{100}$ $\quad P(A) = \frac{1}{10,000}$

$P(\neg B|A) = \frac{1}{100}$

$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)\,P(B|A)}{P(B)}$

$P(B|\neg A) = \frac{1}{100}$

$P(B) = P(B \cap A) + P(B \cap \neg A)$

$= P(A)P(B|A) + P(\neg A)P(B|\neg A)$

$= \frac{1}{10000}\left(\frac{99}{100}\right) + \frac{9999}{10000}\left(\frac{1}{100}\right)$

$P(B) = \frac{10098}{10000000}$

...

$P(A|B) = \frac{1}{10000}\left(\frac{99}{100}\right) \times \frac{10000000}{10098}$

$= \frac{99}{10098}$ Ans

Q13.
In context of Bais-Variance decomposition which of the following statements are true?
Select one or more:
a. High bais implies high vairiance in the out of sample error.
b. High variance implies less bias in the out of sample error.
c. Less bais implies less variance in the out of sample error.
d. Bais-Variance analysis help us to quantify out of sample error.
Answer: B, D
(From L6-Part2 last slide)
(URL: http://www.stat.cmu.edu/~ryantibs/advmethods/notes/errval.pdf)

~~fit on this training set.~~ We'll look at the expected test error, conditional on $X = x$ for some arbitrary input x,

$$\begin{aligned}\mathbb{E}[\text{TestErr}(\hat{r}(x))] &= \mathbb{E}\big[\big(Y - \hat{r}(x)\big)^2 \big| X = x\big] \\ &= \mathbb{E}\big[\big(Y - r(x)\big)^2 \big| X = x\big] + \mathbb{E}\big[\big(r(x) - \hat{r}(x)\big)^2 \big| X = x\big] \\ &= \sigma^2 + \mathbb{E}\big[\big(r(x) - \hat{r}(x)\big)^2\big].\end{aligned}$$

The first term is just a constant, σ^2, and is the *irreducible error* (sometimes referred to as the *Bayes error*). The second term can be further decomposed as

$$\begin{aligned}\mathbb{E}\big[\big(r(x) - \hat{r}(x)\big)^2\big] &= \big(\mathbb{E}[\hat{r}(x)] - r(x)\big)^2 + \mathbb{E}\big[\big(\hat{r}(x) - \mathbb{E}[\hat{r}(x)]\big)^2\big] \\ &= \text{Bias}(\hat{r}(x))^2 + \text{Var}(\hat{r}(x)),\end{aligned}$$

the first term being the squared *estimation bias* or simply *bias*, $\text{Bias}(\hat{r}(x)) = \mathbb{E}[\hat{r}(x)] - r(x)$, and the second term being the *estimation variance* or simply *variance*. Therefore, altogether,

$$\mathbb{E}[\text{TestErr}(\hat{r}(x))] = \sigma^2 + \text{Bias}(\hat{r}(x))^2 + \text{Var}(\hat{r}(x)), \tag{2}$$

which is called the *bias-variance decomposition* or *bias-variance tradeoff*

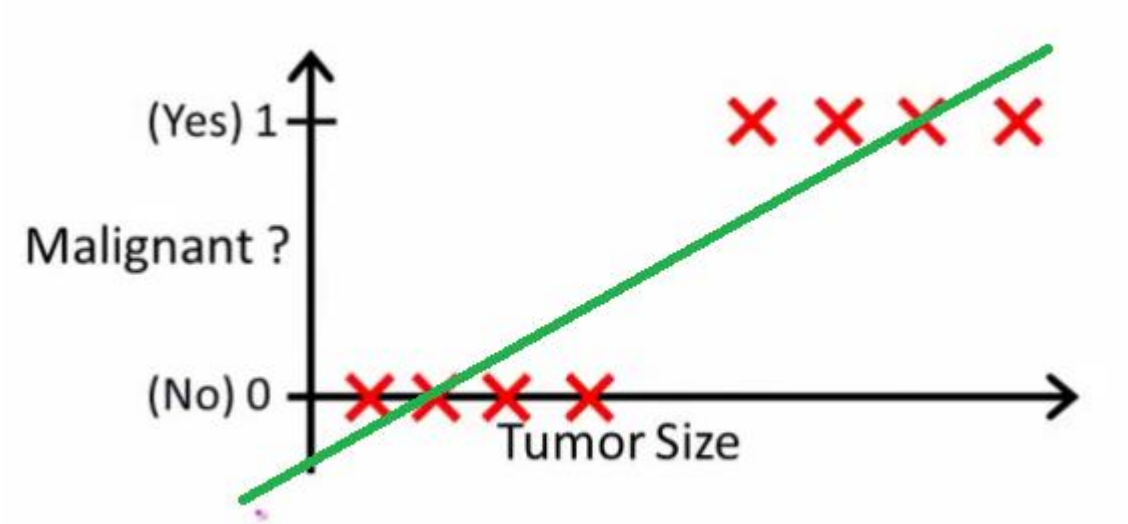
- From the bias-variance tradeoff (2), we can see that even if our prediction is unbiased, i.e., $\mathbb{E}[\hat{r}(x)] = r(x)$, we can still incur a large error if it is highly variable. Meanwhile, even when our prediction is stable and not variable, we can incur a large error if it is badly biased
- There is a tradeoff here, but it need it is really never be one-to-one; i.e., in some cases, it can be worth sacrificying a little bit of bias to gain large decrease in variance, and in other cases, vice versa
- Typical trend: underfitting means high bias and low variance, overfitting means low bias but high variance. E.g., think about k in k-nearest-neighbors regression: relatively speaking, how do the bias and variance behave for small k, and for large k?

Q14.
In context of linear regression, which of the following statements are true?
Select one or more:
a. You can use linear regression for classification.
b. It is not possible to get zero training error, if there are few samples used in training.
c. It is not possible to get zero test error, if there are few samples used in training.
d. You cannot use linear regression for classification.
Answer: A, C
Training error is the error that you get when you run the trained model back on the training data. Remember that this data has already been used to train the model and this necessarily doesn't mean that the model once trained will accurately perform when applied back on the training data itself.

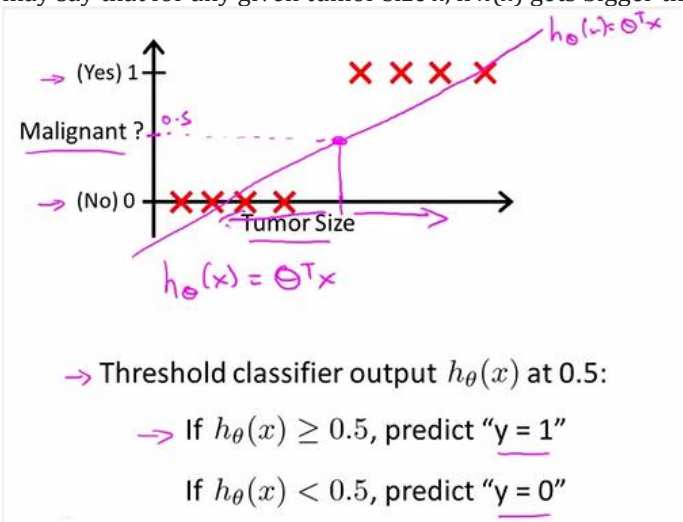
Test error is the error when you get when you run the trained model on a set of data that it has previously never been exposed to. This data is often used to measure the accuracy of the model before it is shipped to production.
……
URL: https://stats.stackexchange.com/questions/22381/why-not-approach-classification-through-regression
QUESTION: Some material I've seen on machine learning said that it's a bad idea to approach a classification problem through regression. But I think it's always possible to do a continuous regression to fit the data and truncate the continuous prediction to yield discrete classifications. So why is it a bad idea?

ANSWER:
"..approach classification problem through regression.." by "regression" I will assume you mean linear regression, and I will compare this approach to the "classification" approach of fitting a logistic regression model.

Before we do this, it is important to clarify the distinction between regression and classification models. Regression models predict a continuous variable, such as rainfall amount or sunlight intensity. They can also predict probabilities, such as the probability that an image contains a cat. A probability-predicting regression model can be used as part of a classifier by imposing a decision rule - for example, if the probability is 50% or more, decide it's a cat.
Logistic regression predicts probabilities, and is therefore a regression algorithm. However, it is commonly described as a classification method in the machine learning literature, because it can be (and is often) used to make classifiers. There are also "true" classification algorithms, such as SVM, which only predict an outcome and do not provide a probability. We won't discuss this kind of algorithm here.
Linear vs. Logistic Regression on Classification Problems
As Andrew Ng explains it, with linear regression you fit a polynomial through the data - say, like on the example below we're fitting a straight line through *{tumor size, tumor type}* sample set:



Above, malignant tumors get 1and non-malignant ones get 0, and the green line is our hypothesis $h(x)$. To make predictions we may say that for any given tumor size x, if $h(x)$ gets bigger than 0.5 we predict malignant tumor, otherwise we predict benign.



Q15.
Which of the following statements are true?
Select one or more:
a. None of these.
b. In multiple classes classification One-versus-the-rest method results in ambiguous regions.
c. Goal in classification is to take an input vector x and to assign it to one of K discrete classes.
d. Discriminant function maps each input x, directly onto the class label.
Answer: B, C, D

Linear Models for Classification

- So far we study regression models.
- This lecture discuss an analogous class of models for solving classification problems.
- Goal in classification is to take an input vector **x** and to assign it to one of K discrete classes C_k
  - ✓ Where $k = 1,2,\ldots, K$
- In the most common scenario
  - ✓ the classes are taken to be disjoint
  - ✓ so that each input is assigned to one and only one class
- The input space is thereby divided into *decision regions* whose boundaries are called *decision boundaries* or *decision surfaces*
- We consider linear models for classification
  - ✓ by which we mean that the decision surfaces are linear functions of the input vector **x**
  - ✓ Hence are defined by $(D-1)$-dimensional hyperplanes within the D-dimensional input space

...

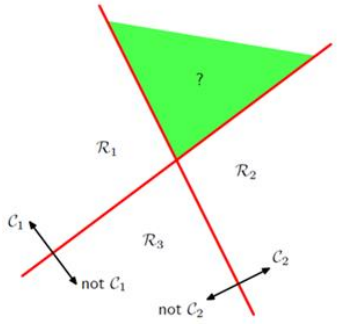
Discriminant Functions

- A **discriminant** is a function that takes an input vector **x** and assigns it to one of K classes, denoted C_k.
- We restrict our attention to *linear discriminants* where the decision surfaces are hyperplanes.
- The simplest representation of a linear discriminant function is obtained by taking a linear function of the input vector so that

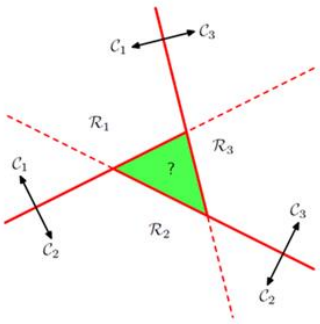
$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

...

Multiple classes classification



One-versus-the-rest method (green region showing the ambiguity)



One-versus-one method

...

BITS WILP Machine Learning Quiz-2 2017-H2

Index

Question 1

Consider the task of building a classifier from random data, where the attribute values are generated randomly irrespective of the class labels. Assume the data set contains records from two classes, "+" and "−." Half of the data set is used for training while the remaining half is used for testing. Suppose there are equal number of "+" and "--" records in test data set and classifier predicts every test record to be positive. What is the expected error rate of the classifier on the test data?

Select one:

- a. Information is not enough.
- b. 0%
- c. 100%
- d. 50%

Feedback

The correct answer is: 50%

Question 2

We built a classifier to detect which gender a name belongs to. We represent each name in term of the following features.

- Number of characters in the name.
- Number of vowels.
- If name contains character y or not. (1 for yes, and 0 for no)

Training data set is as follows:

Akash: Male
Pinky: Female

State which one of the following is true?

Select one or more:

- [x] a. The dimensionality of the data under this feature representation is 3.
- [ ] b. The dimensionality of the data under this feature representation is 2.
- [x] c. The feature vectors of the training name Pinky is [5,1,1].
- [ ] d.

The feature vectors of the training name Akash is [5,2,1].

Feedback

The correct answer is: The dimensionality of the data under this feature representation is 3., The feature vectors of the training name Pinky is [5,1,1].

Question 3

Which of the following statements are true?

Select one or more:

- [x] a. We can use Genetic Algorithm for classification.
- [x] b. Crossover is essential in Genetic Algorithm.
- [ ] c. We cannot use Genetic Algorithm for classification.
- [ ] d. Mutation is essential in Genetic Algorithm.

Feedback

The correct answer is: Crossover is essential in Genetic Algorithm., We can use Genetic Algorithm for classification.

Question 4

Consider the task of building a classifier from random data, where the attribute values are generated randomly irrespective of the class labels. Assume the data set contains records from two classes, "+" and "−." Half of the data set is used for training while the remaining half is used for testing. Suppose there are equal number of "+" and "--" records in test data set and classifier predicts each test record to be positive class with probability 0.8 and negative class with probability 0.2. What is the expected error rate of the classifier on the test data?

Select one:

- a. 100%
- b. 0%
- c. Information is not enough
- d. 50%

Feedback

The correct answer is: 50%

Question 5

Which of the following statements are true about univariate Gaussian Distribution?

Select one or more:

- a. Mean and Mode are different.
- b. Mean and Mode are same.
- c. None of them.
- d. Beta Distribution is prior conjugate.

Feedback

The correct answer is: Mean and Mode are same., Beta Distribution is prior conjugate.

Question 6

Consider the following transformation of the feature space where X = (x1,x2)
$\Phi(x) = (x1^2, \sqrt{2}x1x2, x2^2)$. Which one is the valid kernel for this transformation?

Select one:

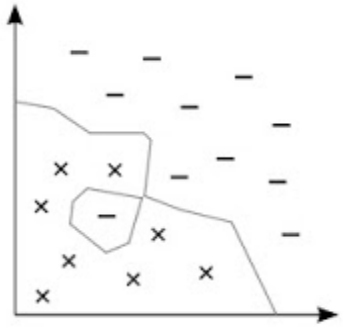
- a. $K(x,z)=(x^Tz)$
- b. $K(x,z)=(x^Tz + 1)^2$
- c. None of them.
- d. $K(x,z)=(x^Tz)^2$

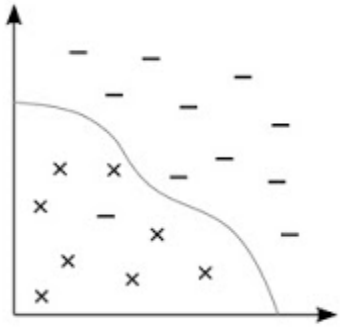
Feedback

The correct answer is:
$K(x,z)=(x^Tz)^2$

Question 7

Figure illustrates decision boundaries for two nearest-neighbor classifiers. Which of the following statements are true?


Decision Boundary 1


Decision Boundary 2

Select one or more:

- [x] a. Decision boundary 1 belongs to 1 nearest neighbor.
- [ ] b. Decision boundary 1 belongs to 3 nearest neighbor.
- [ ] c. Decision boundary 2 belongs to 1 nearest neighbor.
- [x] d. Decision boundary 2 belongs to 3 nearest neighbor.

Feedback

The correct answer is: Decision boundary 1 belongs to 1 nearest neighbor., Decision boundary 2 belongs to 3 nearest neighbor.

Question 8

Which one of the following statements are true?

Select one or more:

- [ ] a. K-NN has slow training time, does not take into account the overall distribution of the data.
- [ ] b. None of them.
- [x] c. Perceptron has slow training time in respect to K-NN.
- [ ] d. K-NN has slow testing time, does take into account the overall distribution of the data.

Feedback

The correct answer is: Perceptron has slow training time in respect to K-NN.

Question 9

Which one of the following statements are true?

Select one or more:

- [x] a.

Neural network use to capture nonlinear relationship between input and output.

- [ ] b.

Perceptron can classify nonlinear separable data accurately.

- [ ] c. All of them.
- [ ] d. Perceptron can only be used to classify two dimensional input.

Feedback

The correct answer is:
Neural network use to capture nonlinear relationship between input and output.

Question 10

Which of the options are true for the given statement.?
Given a set of linearly separable training examples, we train the perceptron algorithm twice, initializing the weights differently for each run. The two training procedures traverse the data points in the same order and are run until convergence.

Select one or more:

- [ ] a. two resulting classifiers have the same performance on the test set
- [x] b. two resulting classifiers have the different performance on the test set
- [ ] c. two resulting classifiers have the different performance on the training set
- [x] d. two resulting classifiers have the same performance on the training set

Feedback

The correct answer is: two resulting classifiers have the same performance on the training set, two resulting classifiers have the different performance on the test set

Question 11

Which of the Boolean function given below is linearly separable?

Select one or more:

- [ ] a. (X XOR Y) AND (X OR Y)
- [x] b. NOT X AND Y
- [x] c. X AND Y AND Z
- [x] d. (X OR Y) AND (X OR Z)

Feedback

The correct answer is: X AND Y AND Z, NOT X AND Y, (X OR Y) AND (X OR Z)

Question 12

State True/False for the following statement. The Support Vectors do not change when we train SVM directly on input and when we map input to the higher dimension and then train SVM.

Select one:

- (•) a. False
- ( ) b. True

Feedback

The correct answer is: False

Question 13

We built a classifier to detect which gender a name belong to. We represent each name in term of the following features.

- Number of characters in the name.
- Number of vowels.
- If name contains character y or not. (1 for yes, and 0 for no)

Training data set is as follows:

Akash: Male

Pinky: Female

Using the kNN algorithm with k = 1 and Euclidean (L2) distance, we want to predict the gender of name = Whisky.

State which of the following statements are true?

Select one or more:

- [ ] a. L2 distance between Akash and Whisky is 3 and classifier predicts Male.
- [ ] b. L2 distance between Akash and Whisky is 2 and classifier predicts Female.
- [x] c. L2 distance between Pinky and Whisky is 1 and classifier predicts Female.
- [ ] d. L2 distance between Pinky and Whisky is 2 and classifier predicts Female.

Feedback

The correct answer is: L2 distance between Pinky and Whisky is 1 and classifier predicts Female.

Question 14

Consider the following training set in 2-dimentional Euclidean space.

Point	Coordinate	Class
X1	(-1, 1)	Negative
X2	(0, 1)	Positive
X3	(0, 2)	Negative
X4	(1, -1)	Positive
X5	(1, 0)	Positive
X6	(1, 2)	Positive
X7	(2, 2)	Negative
X6	(1, 3)	Positive

Which of the following is true?

Select one or more:

☑ a.
If 8NN classifier is consider, point (0,0) belongs to Positive class.

☑ b.
If 5NN classifier is consider, point (0,0) belongs to Positive class.

☑ c. If 2NN classifier is consider, point (0,0) belongs to Positive class.

☑ d.
If 1NN classifier is consider, point (0,0) belongs to Positive class.

Feedback

The correct answer is:
If 8NN classifier is consider, point (0,0) belongs to Positive class.,
If 1NN classifier is consider, point (0,0) belongs to Positive class.,
If 5NN classifier is consider, point (0,0) belongs to Positive class., If 2NN classifier is consider, point (0,0) belongs to Positive class.

Question 15

Which of the following statement are true ?

Select one or more:

☐ a. Genetic algorithm always stuck to the local minimum.

☑ b. Gradient decent method always stuck to the local minimum.

☐ c. Genetic algorithm runs faster than Gradient decent method.

☐ d. All of them.

Feedback

The correct answer is: Gradient decent method always stuck to the local minimum.

BITS WILP Machine Learning Assignment 2017-H1

This programming assignment is on classification. In particular, you have to implement one of the approaches for classification in any of the programming language of your choice.

Following are the sequence of activities involved.

(1) You are given a data set (a classic dataset, iris flower data set) where each observed example has a set of features and has labels. Labels are essential for learning any supervised learning algorithms. Details on the data set is available on https://en.wikipedia.org/wiki/Iris_flower_data_set.
The first task is to understand the data set and understand the classification problem posed on this data. You are free to store this data set in any format for your implementation, however you are not allowed to

modify the data.
(2) Split the data set into training and testing sets. Every third row in the given data set is your testing set and the rest of data makes the training set.
(3) Choose one classifier (from discriminant based / instance based / likelihood based approaches) and implement this classifier, learn the necessary parameters using the training set.
(4) Classify the testing instances with the model built. Compute performance metrics and explain how well the model fit to the given classification problem.

What to Submit?

(1) Implementation Files, along with the input files in a folder named 'implementation'
(2) A text file, detailing the software requirements to run your program, along with the instructions to run this.
(3) A word document, explaining your implementation, and details of activities from (1) to (4). Name this as report.pdf
(4) Zip items (1) to (3) in a file and name it as 'your ID-Name.zip' (All caps) and submit through the documentation

Other details

(1) However, there is no limit on the programming language that you use to implement the assignment, we recommend using C/Java/Python/MatLab.
(2) Recommending algorithms Decision Tree/ LDA/ Logistic Regression/ Naïve Bayes, however, you are free to make a choice.
(3) Ensure that you do not submit any downloaded codes. Write your own implementation. Submit the assignment even you have completed a part of it, and this will be evaluated.
Weightage is 10 % - Number of Days Given to Solve is 10 days – Deadline is 04-04-17

BITS WILP Machine Learning Mid-Sem Exam 2016-H2

Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programmes Division
First Semester 2016-2017
Mid-Semester Test (EC-2 Regular)
Course No. : IS ZC464
Course Title : MACHINE LEARNING
Nature of Exam : Closed Book
Weightage : 35%
Duration : 2 Hours
Date of Exam : 25/09/2016 (FN)
No. of pages: 1; No. of questions: 3
Note:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

Q1. Answer the following questions [4 × 3 = 12]
a) Describe the meaning of *'best hypothesis'* in the context of function approximation in machine learning .
b) What is *Bayes' theorem*? How is it significant in machine learning?
c) Explain *MAP* technique used in learning? Give example.
d) Explain the role of *'error'* in prediction of target value given the test data. Also explain the role of training in prediction.

Q2. Answer the following questions [7 + 7 =14]
a) The chances of children in primary schools in villages dropping (D) their studies are high due to various reasons. The major factors are lack of basic needs of children at home (N) and lack of infrastructure (I) such as school building and availability of teachers who can teach well and motivate student. The statistics collected as the joint probabilities are given in the following table.

	I		~ I	
	N	~N	N	~N
D	0.098	0.022	0.06	0.02
~D	0.018	0.062	0.32	0.4

Use Bayes' theorem to compute the posterior probability P(D | N) using the given joint probabilities. Explain all steps of calculation. [Note: A calculation without the correct expression will not be given credit.]

b) Consider a linear model of the form

$$h(x,W) = w_0 + \sum_{i=1}^{n} w_i \phi_i(x)$$

where $< w_0, w_1, \ldots \ldots w_n >$ is the vector of parameters and is represented as W. The function $\phi_i(x) : i = 1,2,\ldots n$ are the basis functions. Explain the significance of the parameters W in linear regression. Comment on the parameters W in the context of approximating data using a straight line.

Q3. Answer the following questions [5 + 4 = 9]
a) What do you understand by entropy? Calculate the entropy for the following data.

Symbols->	A	B	C	D	E
Probability->	0.20	0.15	0.10	0.35	0.20

b) What is the significance of attribute selection in decision tree based learning? Explain with an appropriate example.