Module 2

INFORMED SEARCH STRATEGIES

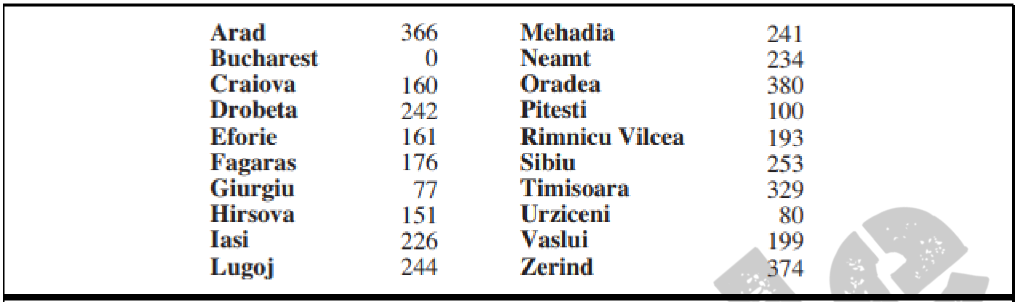
3.5 INFORMED (HEURISTIC) SEARCH STRATEGIES

This section shows how an informed search strategy—one that uses problem-specific knowledge beyond the definition of the problem itself—can find solutions more efficiently than can an uninformed strategy. The general approach we consider is called best-first search. Best-first search is an instance of the general TREE-SEARCH or GRAPH-SEARCH algorithm in which a node is selected for expansion based on an evaluation function, f(n). The evaluation function is construed as a cost estimate, so the node with the lowest evaluation is expanded first. The implementation of best-first graph search is identical to that for uniform-cost search (Figure 3.14), except for the use of f instead of g to order the priority queue. The choice of f determines the search strategy. (For example, as Exercise 3.21 shows, best-first tree search includes depth-first search as a special case.)

Most best-first algorithms include as a component of f a heuristic function, denoted h(n): h(n) = estimated cost of the cheapest path from the state at node n to a goal state.

3.5.1 Greedy best-first search

Greedy best-first search tries to expand the node that is closest to the goal, on the grounds that this is likely to lead to a solution quickly. Thus, it evaluates nodes by using just the heuristic function; that is, f(n) = h(n). Let us see how this works for route-finding problems in Romania; we use the straight line distance heuristic, which we will call hSLD . If the goal is Bucharest, we need to know the straight-line distances to Bucharest, which are shown in Figure 3.22. For example, hSLD (In(Arad)) = 366. Notice that the values of hSLD cannot be computed from the problem description itself. Moreover, it takes a certain amount of experience to know that hSLD is correlated with actual road distances and is, therefore, a useful heuristic. Figure 3.23 shows the progress of a greedy best-first search using hSLD to find a path from Arad to Bucharest. The first node to be expanded from Arad will be Sibiu because it is closer to Bucharest than either Zerind or Timisoara. The next node to be expanded will be Fagaras because it is closest. Fagaras in turn generates Bucharest, which is the goal. For this particular problem, greedy best-first search using hSLD finds a solution without ever expanding a node that is not on the solution path; hence, its search cost is minimal. It is not optimal, however: the path via Sibiu and Fagaras to Bucharest is 32 kilometers longer than the path through Rimnicu Vilcea and Pitesti. This shows why the algorithm is called "greedy"—at each step it tries to get as close to the goal as it can. Greedy best-first tree search is also incomplete even in a finite state space, much like depth-first search. Consider the problem of getting from Iasi to Fagaras. The heuristic suggests that Neamt be expanded first because it is closest to Fagaras, but it is a dead end. The solution is to go first to Vaslui—a step that is actually farther from the goal according to the heuristic—and then to continue to Urziceni, Bucharest, and Fagaras. The algorithm will never find this solution, however, because expanding Neamt puts Iasi back into the frontier, Iasi is closer to Fagaras than Vaslui is, and so Iasi will be expanded again, leading to an infinite loop. (The graph search version is complete in finite spaces, but not in infinite ones.) The worst-case time and space complexity for the tree version is O(bm), where m is the maximum depth of the search space. With a good heuristic function, however, the complexity can be reduced substantially. The amount of the reduction depends on the particular problem and on the quality of the heuristic



Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 3.22 Values of h_{SLD}—straight-line distances to Bucharest.

3.5.2 A* search: Minimizing the total estimated solution cost

The most widely known form of best-first search is called A∗ search (pronounced "A-star search"). It evaluates nodes by combining g(n), the cost to reach the node, and h(n), the cost to get from the node to the goal:

$f(n) = g(n) + h(n)$.

Since g(n) gives the path cost from the start node to node n, and h(n) is the estimated cost of the cheapest path from n to the goal, we have

f(n) = estimated cost of the cheapest solution through n . Thus, if we are trying to find the cheapest solution, a reasonable thing to try first is the node with the lowest value of g(n) + h(n). It turns out that this strategy is more than just reasonable: provided that the heuristic function h(n) satisfies certain conditions, A∗ search is both complete and optimal. The algorithm is identical to UNIFORM-COST-SEARCH except that A∗ uses g + h instead of g.

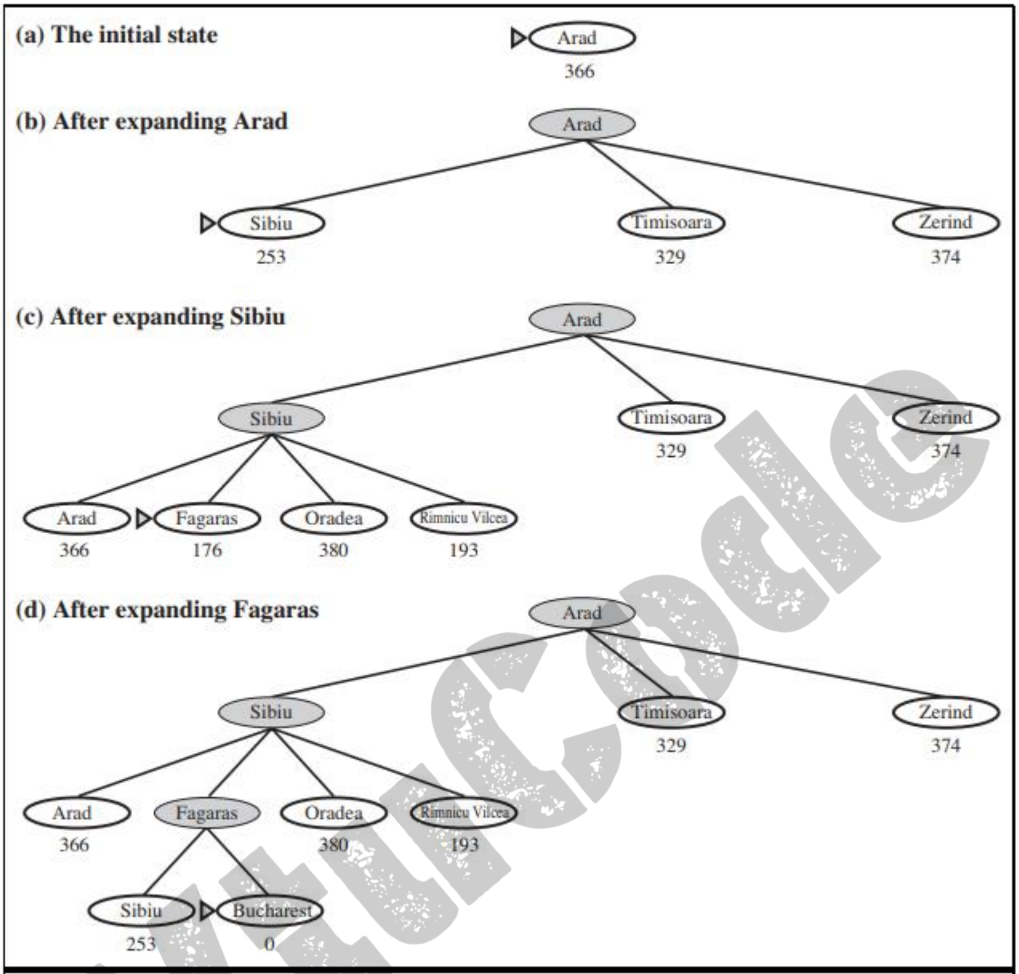


Figure 3.23 Stages in a greedy best-first tree search for Bucharest with the straight-line distance heuristic h_{SLD}. Nodes are labeled with their h-values.

Conditions for optimality: Admissibility and consistency The first condition we require for optimality is that h(n) be an admissible heuristic. An admissible heuristic is one that never overestimates the cost to reach the goal. Because g(n) is the actual cost to reach n along the current path, and f(n) = g(n) + h(n), we have as an immediate consequence that f(n) never overestimates the true cost of a solution along the current path through n. Admissible heuristics are by nature optimistic because they think the cost of solving the problem is less than it actually is. An obvious example of an admissible heuristic is the straight-line distance hSLD that we used in getting to Bucharest. Straight-line distance is admissible because the shortest path between any two points is a straight line, so the straight line cannot be an overestimate. In Figure 3.24, we show the progress of an A∗ tree search for Bucharest. The values of g are computed from the step costs in Figure 3.2, and the values of hSLD are given in Figure 3.22. Notice in particular that Bucharest first appears on the frontier at step (e), but it is not selected for expansion because its f-cost (450) is higher than that of Pitesti (417). Another way to say this is that there might be a solution through Pitesti whose cost is as low as 417, so the algorithm will not settle for a solution that costs 450. A second, slightly stronger condition called consistency (or sometimes monotonicity) is required only for applications of A∗ to graph search.9 A heuristic h(n) is consistent if, for every node n and every successor nof n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to nplus the estimated cost of reaching the goal from n’ :

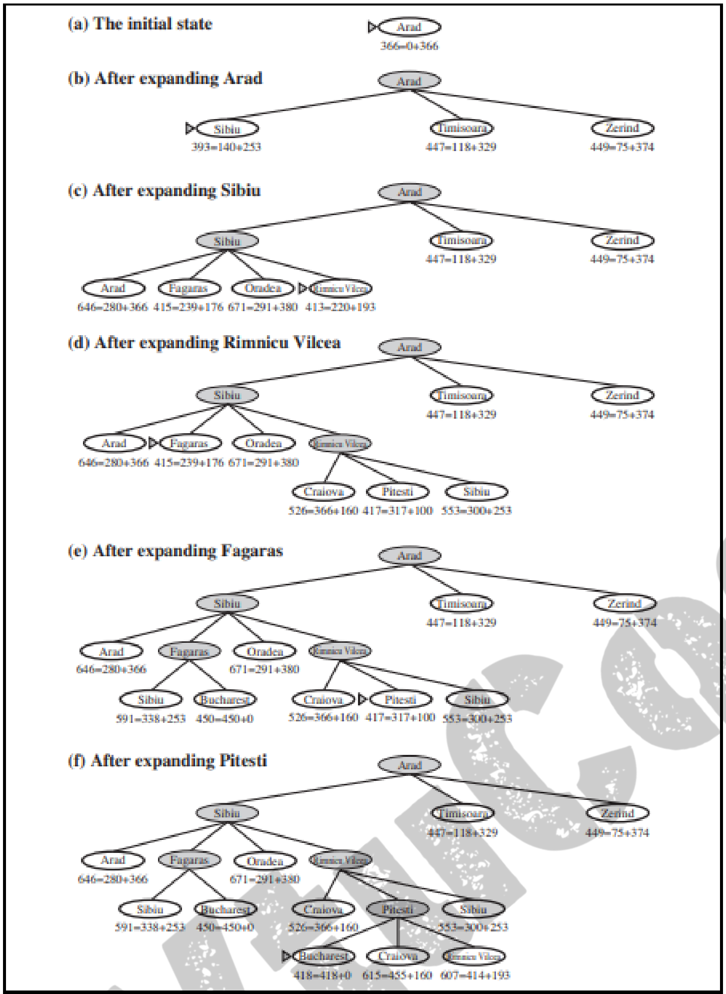
$h(n) \leq c(n, a, n') + h(n')$.

This is a form of the general triangle inequality, which stipulates that each side of a triangle cannot be longer than the sum of the other two sides. Here, the triangle is formed by n, n’, and the goal Gn closest to n. For an admissible heuristic, the inequality makes perfect sense: if there were a route from n to Gn via n’ that was cheaper than h(n), that would violate the property that h(n) is a lower bound on the cost to reach Gn. It is fairly easy to show (Exercise 3.29) that every consistent heuristic is also admissible. Consistency is therefore a stricter requirement than admissibility, but one has to work quite hard to concoct heuristics that are admissible but not consistent. All the admissible heuristics we discuss in this chapter are also consistent. Consider, for example, hSLD . We know that the general triangle inequality is satisfied when each side is measured by the straight-line distance and that the straight-line distance between n and n’ is no greater than c(n, a, n’). Hence, hSLD is a consistent heuristic.

Optimality of A* As we mentioned earlier, A∗ has the following properties: the tree-search version of A∗ is optimal if h(n) is admissible, while the graph-search version is optimal if h(n) is consistent. We show the second of these two claims since it is more useful. The argument essentially mirrors the argument for the optimality of uniform-cost search, with g replaced by f—just as in the A∗ algorithm itself. The first step is to establish the following: if h(n) is consistent, then the values of f(n) along any path are nondecreasing. The proof follows directly from the definition of consistency. Suppose n is a successor of n; then g(n’) = g(n) + c(n, a, n’) for some action a, and we have

$f(n') = g(n') + h(n') = g(n) + c(n, a, n') + h(n') \geq g(n) + h(n) = f(n)$.

The next step is to prove that whenever A∗ selects a node n for expansion, the optimal path to that node has been found. Were this not the case, there would have to be another frontier node n on the optimal path from the start node to n, by the graph separation property of



(a) The initial state

(b) After expanding Arad

(c) After expanding Sibiu

(d) After expanding Rimnicu Vilcea

(e) After expanding Fagaras

(f) After expanding Pitesti

Figure 3.24 Stages in an A* search for Bucharest. Nodes are labeled with $f = g + h$. The h values are the straight-line distances to Bucharest taken from Figure 3.22.

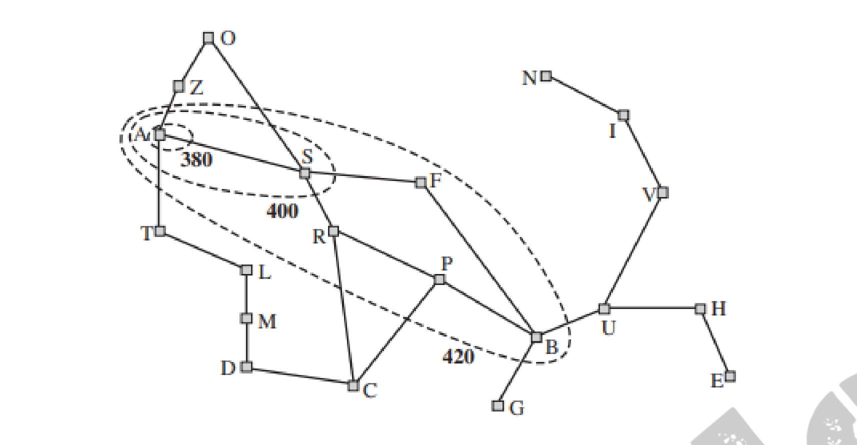
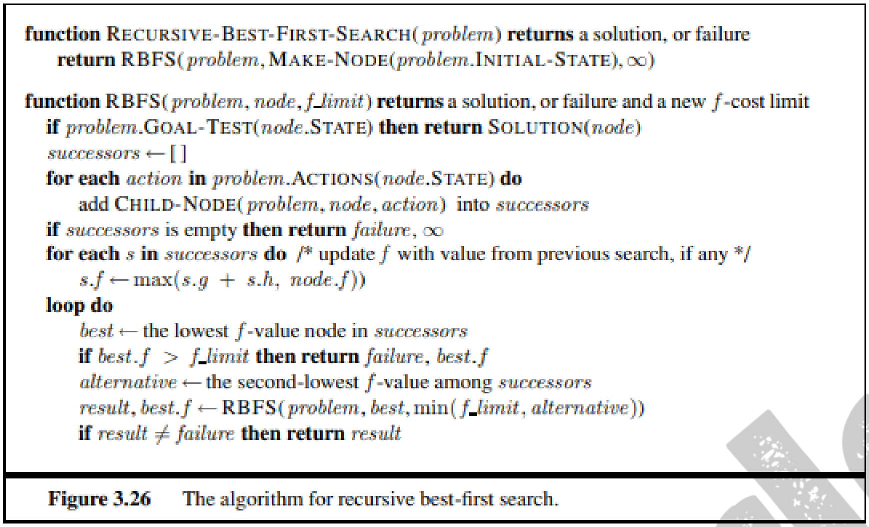


Figure 3.25 Map of Romania showing contours at $f = 380$, $f = 400$, and $f = 420$, with Arad as the start state. Nodes inside a given contour have f-costs less than or equal to the contour value.

Figure 3.9; because f is nondecreasing along any path, nwould have lower f-cost than n and would have been selected first. From the two preceding observations, it follows that the sequence of nodes expanded by A∗ using GRAPH-SEARCH is in nondecreasing order of f(n). Hence, the first goal node selected for expansion must be an optimal solution because f is the true cost for goal nodes (which have h = 0) and all later goal nodes will be at least as expensive. The fact that f-costs are nondecreasing along any path also means that we can draw contours in the state space, just like the contours in a topographic map. Figure 3.25 shows an example. Inside the contour labeled 400, all nodes have f(n) less than or equal to 400, and so on. Then, because A∗ expands the frontier node of lowest f-cost, we can see that an A∗ search fans out from the start node, adding nodes in concentric bands of increasing f-cost. With uniform-cost search (A∗ search using h(n)=0), the bands will be "circular" around the start state. With more accurate heuristics, the bands will stretch toward the goal state and become more narrowly focused around the optimal path. If C∗ is the cost of the optimal solution path, then we can say the following:

- A∗ expands all nodes with f(n) < C∗.
- A∗ might then expand some of the nodes right on the "goal contour" (where f(n) = C∗) before selecting a goal node. Completeness requires that there be only finitely many nodes with cost less than or equal to C∗, a condition that is true if all step costs exceed some finite and if b is finite. Notice that A∗ expands no nodes with f(n) > C∗—for example, Timisoara is not expanded in Figure 3.24 even though it is a child of the root. We say that the subtree below

Timisoara is pruned; because hSLD is admissible, the algorithm can safely ignore this subtree while still guaranteeing optimality. The concept of pruning—eliminating possibilities from consideration without having to examine them—is important for many areas of AI. One final observation is that among optimal algorithms of this type—algorithms that extend search paths from the root and use the same heuristic information—A∗ is optimally efficient for any given consistent heuristic. That is, no other optimal algorithm is guaran- teed to expand fewer nodes than A∗ (except possibly through tie-breaking among nodes with $f(n) = C*$). This is because any algorithm that does not expand all nodes with $f(n) < C*$ runs the risk of missing the optimal solution. That A∗ search is complete, optimal, and optimally efficient among all such algorithms is rather satisfying. Unfortunately, it does not mean that A∗ is the answer to all our searching needs. The catch is that, for most problems, the number of states within the goal contour search space is still exponential in the length of the solution. The details of the analysis are beyond the scope of this book, but the basic results are as follows. For problems with constant step costs, the growth in run time as a function of the optimal solution depth d is analyzed in terms of the the absolute error or the relative error of the heuristic. The absolute error is defined as $\Delta \equiv h* - h$, where h∗ is the actual cost of getting from the root to the goal, and the relative error is defined as $\equiv (h* - h)/h*$. The complexity results depend very strongly on the assumptions made about the state space. The simplest model studied is a state space that has a single goal and is essentially a tree with reversible actions. (The 8-puzzle satisfies the first and third of these assumptions.) In this case, the time complexity of A∗ is exponential in the maximum absolute error, that is, O(b^Δ). For constant step costs, we can write this as O(b^d), where d is the solution depth. For almost all heuristics in practical use, the absolute error is at least proportional to the path cost h∗, so is constant or growing and the time complexity is exponential in d. We can also see the effect of a more accurate heuristic: O(b^d) = O((b^)^d), so the effective branching factor (defined more formally in the next section) is b^. When the state space has many goal states—particularly near-optimal goal states—the search process can be led astray from the optimal path and there is an extra cost proportional to the number of goals whose cost is within a factor of the optimal cost. Finally, in the general case of a graph, the situation is even worse. There can be exponentially many states with $f(n) < C*$ even if the absolute error is bounded by a constant. For example, consider a version of the vacuum world where the agent can clean up any square for unit cost without even having to visit it: in that case, squares can be cleaned in any order. With N initially dirty squares, there are 2N states where some subset has been cleaned and all of them are on an optimal solution path—and hence satisfy $f(n) < C*$—even if the heuristic has an error of 1. The complexity of A∗ often makes it impractical to insist on finding an optimal solution. One can use variants of A∗ that find suboptimal solutions quickly, or one can sometimes design heuristics that are more accurate but not strictly admissible. In any case, the use of a good heuristic still provides enormous savings compared to the use of an uninformed search. In Section 3.6, we look at the question of designing good heuristics. Computation time is not, however, A∗'s main drawback. Because it keeps all generated nodes in memory (as do all GRAPH-SEARCH algorithms), A∗ usually runs out of space long before it runs out of time. For this reason, A∗ is not practical for many large-scale problems. There are, however, algorithms that overcome the space problem without sacrificing optimality or completeness, at a small cost in execution time. We discuss these next.



```
function RECURSIVE-BEST-FIRST-SEARCH(problem) returns a solution, or failure
    return RBFS(problem, MAKE-NODE(problem.INITIAL-STATE), ∞)

function RBFS(problem, node, f_limit) returns a solution, or failure and a new f-cost limit
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  successors ← [ ]
  for each action in problem.ACTIONS(node.STATE) do
      add CHILD-NODE(problem, node, action) into successors
  if successors is empty then return failure, ∞
  for each s in successors do  /* update f with value from previous search, if any */
      s.f ← max(s.g + s.h, node.f))
  loop do
      best ← the lowest f-value node in successors
      if best.f > f_limit then return failure, best.f
      alternative ← the second-lowest f-value among successors
      result, best.f ← RBFS(problem, best, min(f_limit, alternative))
      if result ≠ failure then return result
```

Figure 3.26 The algorithm for recursive best-first search.

3.6 HEURISTIC FUNCTIONS

In this section, we look at heuristics for the 8-puzzle, in order to shed light on the nature of heuristics in general. The 8-puzzle was one of the earliest heuristic search problems. As mentioned in Section 3.2, the object of the puzzle is to slide the tiles horizontally or vertically into the empty space until the configuration matches the goal configuration (Figure 3.28). The average solution cost for a randomly generated 8-puzzle instance is about 22 steps. The branching factor is about 3. (When the empty tile is in the middle, four moves are possible; when it is in a corner, two; and when it is along an edge, three.) This means that an exhaustive tree search to depth 22 would look at about 3^22 ≈ 3.1 × 10^10 states. A graph search would cut this down by a factor of about 170,000 because only 9!/2 = 181, 440 distinct states are reachable. (See Exercise 3.4.) This is a manageable number, but the corresponding number for the 15-puzzle is roughly 10^13, so the next order of business is to find a good heuristic function. If we want to find the shortest solutions by using A∗, we need a heuristic function that never overestimates the number of steps to the goal. There is a long history of such heuristics for the 15-puzzle; here are two commonly used candidates:

- h1 = the number of misplaced tiles. For Figure 3.28, all of the eight tiles are out of position, so the start state would have h1 = 8. h1 is an admissible heuristic because it is clear that any tile that is out of place must be moved at least once.
- h2 = the sum of the distances of the tiles from their goal positions. Because tiles cannot move along diagonals, the distance we will count is the sum of the horizontal and vertical distances. This is sometimes called the city block distance or Manhattan distance. h2 is also admissible because all any move can do is move one tile one step MANHATTAN DISTANCE closer to the goal. Tiles 1 to 8 in the start state give a Manhattan distance of h2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18 . As expected, neither of these overestimates the true solution cost, which is 26.

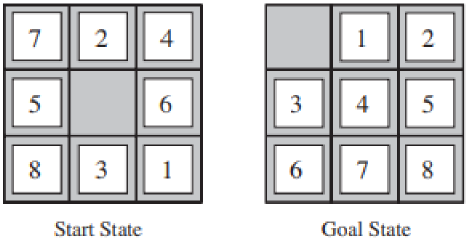


Figure 3.28 A typical instance of the 8-puzzle. The solution is 26 steps long.

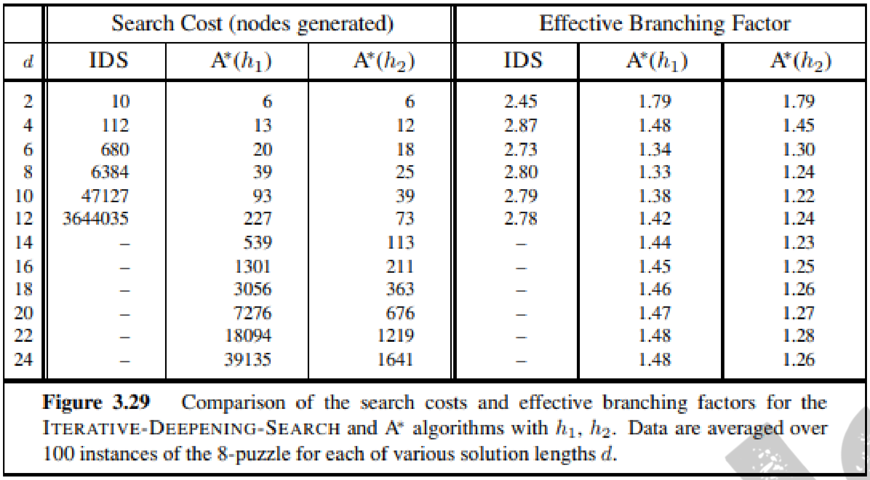
3.6.1 The effect of heuristic accuracy on performance

One way to characterize the quality of a heuristic is the effective branching factor b∗. If the total number of nodes generated by A∗ for a particular problem is N and the solution depth is d, then b∗ is the branching factor that a uniform tree of depth d would have to have in order to contain N + 1 nodes. Thus,

$$N + 1 = 1 + b^* + (b^*)^2 + \cdots + (b^*)^d .$$

For example, if A∗ finds a solution at depth 5 using 52 nodes, then the effective branching factor is 1.92. The effective branching factor can vary across problem instances, but usually it is fairly constant for sufficiently hard problems. (The existence of an effective branching factor follows from the result, mentioned earlier, that the number of nodes expanded by A∗ grows exponentially with solution depth.) Therefore, experimental measurements of b∗ on a small set of problems can provide a good guide to the heuristic's overall usefulness. A welldesigned heuristic would have a value of b∗ close to 1, allowing fairly large problems to be solved at reasonable computational cost

To test the heuristic functions h1 and h2, we generated 1200 random problems with solution lengths from 2 to 24 (100 for each even number) and solved them with iterative deepening search and with A∗ tree search using both h1 and h2. Figure 3.29 gives the average number of nodes generated by each strategy and the effective branching factor. The results suggest that h2 is better than h1, and is far better than using iterative deepening search. Even for small problems with d = 12, A∗ with h2 is 50,000 times more efficient than uninformed iterative deepening search



	Search Cost (nodes generated)			Effective Branching Factor		
d	IDS	A*(h_1)	A*(h_2)	IDS	A*(h_1)	A*(h_2)
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	–	539	113	–	1.44	1.23
16	–	1301	211	–	1.45	1.25
18	–	3056	363	–	1.46	1.26
20	–	7276	676	–	1.47	1.27
22	–	18094	1219	–	1.48	1.28
24	–	39135	1641	–	1.48	1.26

Figure 3.29 Comparison of the search costs and effective branching factors for the ITERATIVE-DEEPENING-SEARCH and A* algorithms with h_1, h_2. Data are averaged over 100 instances of the 8-puzzle for each of various solution lengths d.

One might ask whether h2 is always better than h1. The answer is "Essentially, yes." It is easy to see from the definitions of the two heuristics that, for any node n, h2(n) ≥ h1(n). We thus say that h2 dominates h1. Domination translates directly into efficiency: A∗ using h2 will never expand more nodes than A∗ using h1 (except possibly for some nodes with f(n) = C∗). The argument is simple. Recall the observation on page 97 that every node with f(n) < C∗ will surely be expanded. This is the same as saying that every node with h(n) < C∗ − g(n) will surely be expanded. But because h2 is at least as big as h1 for all nodes, every node that is surely expanded by A∗ search with h2 will also surely be expanded with h1, and h1 might cause other nodes to be expanded as well. Hence, it is generally better to use a heuristic function with higher values, provided it is consistent and that the computation time for the heuristic is not too long.

3.6.2 Generating admissible heuristics from relaxed problems

We have seen that both h1 (misplaced tiles) and h2 (Manhattan distance) are fairly good heuristics for the 8-puzzle and that h2 is better. How might one have come up with h2? Is it possible for a computer to invent such a heuristic mechanically? h1 and h2 are estimates of the remaining path length for the 8-puzzle, but they are also perfectly accurate path lengths for simplified versions of the puzzle. If the rules of the puzzle were changed so that a tile could move anywhere instead of just to the adjacent empty square, then h1 would give the exact number of steps in the shortest solution. Similarly, if a tile could move one square in any direction, even onto an occupied square, then h2 would give the exact number of steps in the shortest solution. A problem with fewer restrictions on the actions is called a relaxed problem. The state-space graph of the relaxed problem is a supergraph of the original state space because the removal of restrictions creates added edges in the graph. Because the relaxed problem adds edges to the state space, any optimal solution in the original problem is, by definition, also a solution in the relaxed problem; but the relaxed problem may have better solutions if the added edges provide short cuts. Hence, the cost of an optimal solution to a relaxed problem is an admissible heuristic for the original problem. Furthermore, because the derived heuristic is an exact cost for the relaxed problem, it must obey the triangle inequality and is therefore consistent (see page 95). If a problem definition is written down in a formal language, it is possible to construct relaxed problems automatically.11 For example, if the 8-puzzle actions are described as A tile can move from square A to square B if A is horizontally or vertically adjacent to B and B is blank, we can generate three relaxed problems by removing one or both of the conditions:

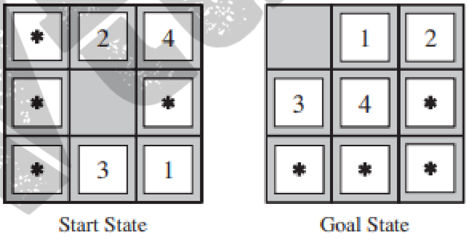
(a) A tile can move from square A to square B if A is adjacent to B.

(b) A tile can move from square A to square B if B is blank.

(c) A tile can move from square A to square B.

From (a), we can derive h2 (Manhattan distance). The reasoning is that h2 would be the proper score if we moved each tile in turn to its destination. The heuristic derived from (b) is discussed in Exercise 3.31. From (c), we can derive h1 (misplaced tiles) because it would be the proper score if tiles could move to their intended destination in one step. Notice that it is crucial that the relaxed problems generated by this technique can be solved essentially without search, because the relaxed rules allow the problem to be decomposed into eight independent subproblems. If the relaxed problem is hard to solve, then the values of the corresponding heuristic will be expensive to obtain.12 A program called ABSOLVER can generate heuristics automatically from problem definitions, using the "relaxed problem" method and various other techniques (Prieditis, 1993). ABSOLVER generated a new heuristic for the 8-puzzle that was better than any preexisting heuristic and found the first useful heuristic for the famous Rubik's Cube puzzle. One problem with generating new heuristic functions is that one often fails to get a single "clearly best" heuristic. If a collection of admissible heuristics h1 ...hm is available for a problem and none of them dominates any of the others, which should we choose? As it turns out, we need not make a choice. We can have the best of all worlds, by defining

h(n) = max{h1(n),...,hm(n)} .



*	2	4
*		*
*	3	1

	1	2
3	4	*
*	*	*

Figure 3.30 A subproblem of the 8-puzzle instance given in Figure 3.28. The task is to get tiles 1, 2, 3, and 4 into their correct positions, without worrying about what happens to the other tiles.

This composite heuristic uses whichever function is most accurate on the node in question. Because the component heuristics are admissible, h is admissible; it is also easy to prove that h is consistent. Furthermore, h dominates all of its component heuristics.

3.6.3 Generating admissible heuristics from subproblems: Pattern databases

Admissible heuristics can also be derived from the solution cost of a subproblem of a given problem. For example, Figure 3.30 shows a subproblem of the 8-puzzle instance in Figure

3.28. The subproblem involves getting tiles 1, 2, 3, 4 into their correct positions. Clearly, the cost of the optimal solution of this subproblem is a lower bound on the cost of the complete problem. It turns out to be more accurate than Manhattan distance in some cases. The idea behind pattern databases is to store these exact solution costs for every possible subproblem instance—in our example, every possible configuration of the four tiles and the blank. (The locations of the other four tiles are irrelevant for the purposes of solving the subproblem, but moves of those tiles do count toward the cost.) Then we compute an admissible heuristic hDB for each complete state encountered during a search simply by looking up the corresponding subproblem configuration in the database. The database itself is constructed by searching back13 from the goal and recording the cost of each new pattern encountered; the expense of this search is amortized over many subsequent problem instances. The choice of 1-2-3-4 is fairly arbitrary; we could also construct databases for 5-6-7-8, for 2-4-6-8, and so on. Each database yields an admissible heuristic, and these heuristics can be combined, as explained earlier, by taking the maximum value. A combined heuristic of this kind is much more accurate than the Manhattan distance; the number of nodes generated when solving random 15-puzzles can be reduced by a factor of 1000. One might wonder whether the heuristics obtained from the 1-2-3-4 database and the 5-6-7-8 could be added, since the two subproblems seem not to overlap. Would this still give an admissible heuristic? The answer is no, because the solutions of the 1-2-3-4 subproblem and the 5-6-7-8 subproblem for a given state will almost certainly share some moves—it is unlikely that 1-2-3-4 can be moved into place without touching 5-6-7-8, and vice versa. But what if we don't count those moves? That is, we record not the total cost of solving the 1-2- 3-4 subproblem, but just the number of moves involving 1-2-3-4. Then it is easy to see that the sum of the two costs is still a lower bound on the cost of solving the entire problem. This is the idea behind disjoint pattern databases. With such databases, it is possible to solve random 15-puzzles in a few milliseconds—the number of nodes generated is reduced by a factor of 10,000 compared with the use of Manhattan distance. For 24-puzzles, a speedup of roughly a factor of a million can be obtained. Disjoint pattern databases work for sliding-tile puzzles because the problem can be divided up in such a way that each move affects only one subproblem—because only one tile is moved at a time. For a problem such as Rubik's Cube, this kind of subdivision is difficult because each move affects 8 or 9 of the 26 cubies. More general ways of defining additive, admissible heuristics have been proposed that do apply to Rubik's cube (Yang et al., 2008), but they have not yielded a heuristic better than the best nonadditive heuristic for the problem.

3.6.4 Learning heuristics from experience

A heuristic function h(n) is supposed to estimate the cost of a solution beginning from the state at node n. How could an agent construct such a function? One solution was given in the preceding sections—namely, to devise relaxed problems for which an optimal solution can be found easily. Another solution is to learn from experience. "Experience" here means solving lots of 8-puzzles, for instance. Each optimal solution to an 8-puzzle problem provides examples from which h(n) can be learned. Each example consists of a state from the solution path and the actual cost of the solution from that point. From these examples,

a learning algorithm can be used to construct a function h(n) that can (with luck) predict solution costs for other states that arise during search. Techniques for doing just this using neural nets, decision trees, and other methods are demonstrated in Chapter 18. (The reinforcement learning methods described in Chapter 21 are also applicable.) Inductive learning methods work best when supplied with features of a state that are relevant to predicting the state’s value, rather than with just the raw state description. For example, the feature “number of misplaced tiles” might be helpful in predicting the actual distance of a state from the goal. Let’s call this feature x1(n). We could take 100 randomly generated 8-puzzle configurations and gather statistics on their actual solution costs. We might find that when x1(n) is 5, the average solution cost is around 14, and so on. Given these data, the value of x1 can be used to predict h(n). Of course, we can use several features. A second feature x2(n) might be “number of pairs of adjacent tiles that are not adjacent in the goal state.” How should x1(n) and x2(n) be combined to predict h(n)? A common approach is to use a linear combination:
$h(n) = c1x1(n) + c2x2(n)$.
The constants c1 and c2 are adjusted to give the best fit to the actual data on solution costs. One expects both c1 and c2 to be positive because misplaced tiles and incorrect adjacent pairs make the problem harder to solve. Notice that this heuristic does satisfy the condition that h(n)=0 for goal states, but it is not necessarily admissible or consistent.

INTRODUCTION TO MACHINE LEARNING

1.1 NEED FOR MACHINE LEARNING

Business organizations use huge amount of data for their daily activities. They have now started to use the latest technology, machinelearning, to manage the data.

Machine learning has become so popular because of three reasons:

1. High volume of available data to manage: Big companies such as Facebook, Twitter, and YouTube generate huge amount of data that grows at a phenomenal rate. It is estimated that the data approximately gets doubled every year.
2. Second reason is that the cost of storage has reduced. The hardware cost has also dropped.Therefore, it is easier now to capture, process, store, distribute, and transmit the digital information.
3. Third reason for popularity of machine learning is the availability of complex algorithms now. Especially with the advent of deep learning, many algorithms are available for machine learning.

let us establish these terms - data, information, knowledge, intelligence, and wisdom using a knowledge pyramid as shown in Figure 1.1.

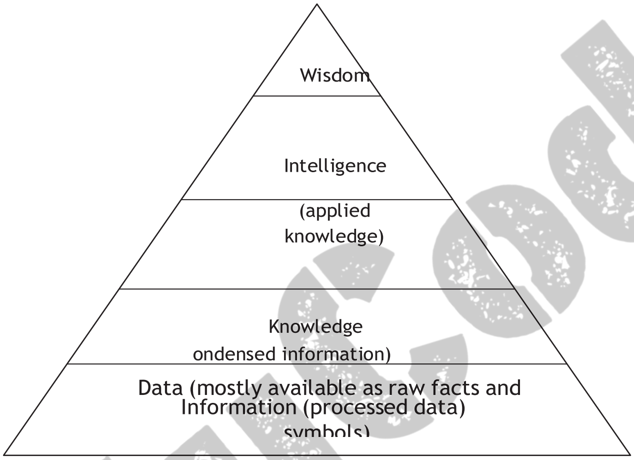


Figure 1.1: The Knowledge Pyramid

- All facts are data. Data can be numbers or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data with data sources such as flat files, databases, or data warehouses in different storage formats.
- Processed data is called information. This includes patterns, associations, or relationships among data. For example, sales data can be analyzed to extract information like which is the fast selling product.
- Condensed information is called knowledge. For example, the historical patterns and future trends obtained in the above sales data can be called knowledge. Unless knowledge is extracted, data is of no use. Similarly, knowledge is not useful unless it is put into action.
- Intelligence is the applied knowledge for actions. An actionable form of knowledge is called intelligence. Computer systems have been successful till this stage.
- The ultimate objective of knowledge pyramid is wisdom that represents the maturity of mind that is, so far, exhibited only by humans.

The objective of machine learning is to process these archival data for organizations to take better decisions to design new products, improve the business processes, and to develop effective decision support systems.

1.2 MACHINE LEARNING EXPLAINED

Machine learning is an important sub-branch of Artificial Intelligence (AI). A frequently quoted definition of machine learning was by Arthur Samuel, one of the pioneers of Artificial Intelligence.He stated that "***Machine learning is the field of study that gives the computers ability to learn without being explicitly programmed***."

The key to this definition is that the systems should learn by itself without explicit programming.How is it possible? It is widely known that to perform a computation, one needs to write programsthat teach the computers how to do that computation.

In conventional programming, after understanding the problem, a detailed design of the program such as a flowchart or an algorithm needs to be created and converted into programs using a suitable programming language. This approach could be difficult for many real-world problems such as puzzles, games, and complex image recognition applications. Initially, artificial intelligence aims to understand these problems and develop general purpose rules manually. Then, these rules are formulated into logic and implemented in a program to create intelligent systems. This idea of developing intelligent systems by using logic and reasoning by converting anexpert's knowledge into a set of rules and programs is called an expert system. An expert system like MYCIN was designed for medical diagnosis after converting the expert knowledge of many doctors into a system. However, this approach did not progress much as programs lacked real intelligence. The word MYCIN is derived from the fact that most of the antibiotics' names end with'mycin'.

The above approach was impractical in many domains as programs still depended on humanexpertise and hence did not truly exhibit intelligence. Then, the momentum shifted to machine learning in the form of data driven systems. The focus of AI is to develop intelligent systems by using data-driven approach, where data is used as an input to develop intelligent models. The models can then be used to predict new inputs. Thus, the aim of machine learning is to learn a model or set of rules from the given dataset automatically so that it can predict the unknown datacorrectly.

As humans take decisions based on an experience, computers make models based on extracted patterns in the input data and then use these data-filled models for prediction and to take decisions. For computers, the learnt model is equivalent to human experience. This is shown in Figure 1.2.

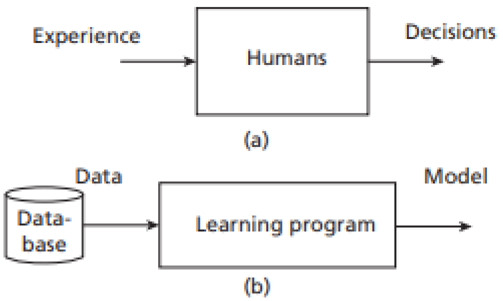


Figure 1.2: (a) A Learning System for Humans (b) A Learning Systemfor Machine Learning

Often, the quality of data determines the quality of experience and, therefore, the quality ofthe learning system. In statistical learning, the relationship between the input *x* and output *y* is modeled as a function in the form *y* = *f*(*x*). Here, f is the learning function that maps the input *x*to output *y*. Learning of function *f* is the crucial aspect of forming a model in statistical learning.In machine learning, this is simply called mapping of input to output.

The learning program summarizes the raw data in a model. Formally stated, a model is anexplicit description of patterns within the data in the form of:

1. Mathematical equation
2. Relational diagrams like trees/graphs
3. Logical if/else rules, or
4. Groupings called clusters

In summary, a model can be a formula, procedure or representation that can generate data decisions. The difference between pattern and model is that the former is local and applicable onlyto certain attributes but the latter is global and fits the entire dataset. For example, a model can behelpful to examine whether a given email is spam or not. The point is that the model is generated automatically from the given data.

Another pioneer of AI, Tom Mitchell's definition of machine learning states that, "*A computer program is said to learn from experience E, with respect to task T and some performance measure P,if its performance on T measured by P improves with experience E*." The important components of this definition are experience *E*, task *T*, and performance measure *P*.

For example, the task *T* could be detecting an object in an image. The machine can gain the knowledge of object using training dataset of thousands of images. This is called experience *E*.So, the focus is to use this experience *E* for this task of object detection *T*. The ability of the systemto detect the object is measured by performance measures like precision and recall. Based on the performance measures, course correction can be done to improve the performance of the system.

Models of computer systems are equivalent to human experience. Experience is based on data. Humans gain experience by various means. They gain knowledge by rote learning. They observe others and imitate it. Humans gain a lot of knowledge from teachers and books. We learn many things by trial and error. Once the knowledge is gained, when a new problem is encountered, humans search for similar past situations and then formulate the heuristics and use that for prediction. But, in systems, experience is gathered by these steps:

1. Collection of data

2. Once data is gathered, abstract concepts are formed out of that data. Abstraction is used to generate concepts. This is equivalent to humans' idea of objects, for example, we have some idea about how an elephant looks like.
3. Generalization converts the abstraction into an actionable form of intelligence. It can be viewed as ordering of all possible concepts. So, generalization involves ranking of concepts, inferencing from them and formation of heuristics, an actionable aspect of intelligence. Heuristics are educated guesses for all tasks. For example, if one runs or encounters a danger, it is the resultant of human experience or his heuristics formation.In machines, it happens the same way.
4. Heuristics normally works! But, occasionally, it may fail too. It is not the faultof heuristics as it is just a 'rule of thumb'. The course correction is done by taking evaluation measures. Evaluation checks the thoroughness of the models and to-do course correction, if necessary, to generate better formulations.

1.3 MACHINE LEARNING IN RELATION TO OTHER FIELDS

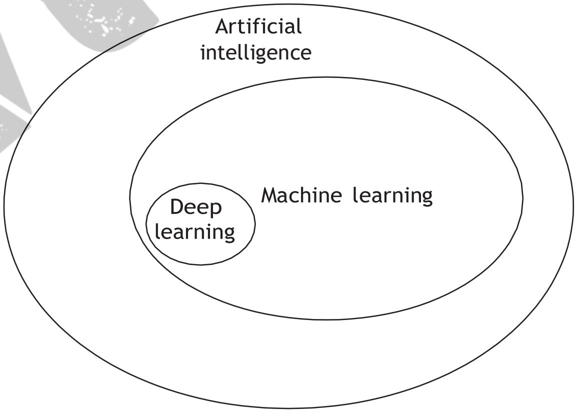
Machine learning uses the concepts of Artificial Intelligence, Data Science, and Statistics primarily.It is the resultant of combined ideas of diverse fields.

1.3.1 Machine Learning and Artificial Intelligence

Machine learning is an important branch of AI, which is a much broader subject. The aim of AI is to develop intelligent agents. An agent can be a robot, humans, or any autonomous systems. Initially, the idea of AI was ambitious, that is, to develop intelligent systems like human beings. The focus was on logic and logical inferences. It had seen many ups and downs. These down periods were called AI winters.

The resurgence in AI happened due to development of data driven systems. The aim is to find relations and regularities present in the data. Machine learning is the subbranch of AI, whose aimis to extract the patterns for prediction. It is a broad field that includes learning from examples andother areas like reinforcement learning. The relationship of AI and machine learning is shown in Figure 1.3. The model can take an unknown instance and generate results.

Figure 1.3: Relationship of AI with Machine Learning



Deep learning is a subbranch of machine learning. In deep learning, the models are constructedusing neural network technology. Neural networks are based on the human neuron models. Many neurons form a network connected with the activation functions that trigger further neurons to perform tasks.

1.3.2 Machine Learning, Data Science, Data Mining, and Data Analytics

Data science is an ‘Umbrella’ term that encompasses many fields. Machine learning starts with data. Therefore, data science and machine learning are interlinked. Machine learning is a branch of data science. Data science deals with gathering of data for analysis. It is a broad field that includes:

Big Data Data science concerns about collection of data. Big data is a field of data science that deals with data’s following characteristics:

1. Volume: Huge amount of data is generated by big companies like Facebook, Twitter, YouTube.
2. Variety: Data is available in variety of forms like images, videos, and in different formats.
3. Velocity: It refers to the speed at which the data is generated and processed.

Big data is used by many machine learning algorithms for applications such as language trans-lation and image recognition. Big data influences the growth of subjects like Deep learning. Deep learning is a branch of machine learning that deals with constructing models using neural networks.

Data Mining Data mining’s original genesis is in the business. Like while mining the earth one gets into precious resources, it is often believed that unearthing of the data produces hidden infor- mation that otherwise would have eluded the attention of the management. Nowadays, many consider that data mining and machine learning are same. There is no difference between these fields except that data mining aims to extract the hidden patterns that are present in the data, whereas, machine learning aims to use it for prediction.

Data Analytics Another branch of data science is data analytics. It aims to extract useful knowledge from crude data. There are different types of analytics. Predictive data analytics is used for making predictions. Machine learning is closely related to this branch of analytics and shares almost all algorithms.

Pattern Recognition It is an engineering field. It uses machine learning algorithms to extract the features for pattern analysis and pattern classification. One can view pattern recognition as a specific application of machine learning.

These relations are summarized in Figure 1.4.

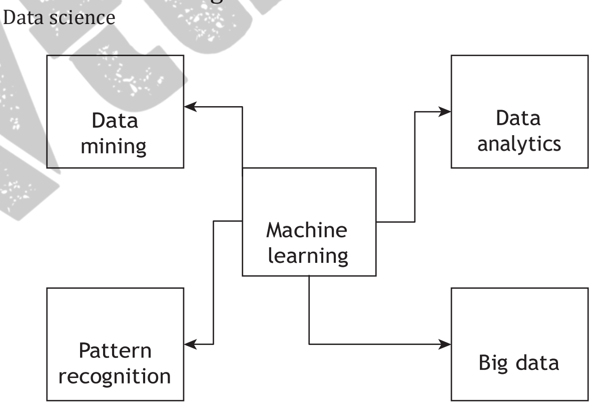


Figure 1.4: Relationship of Machine Learning with Other Major Fields

1.3.3 Machine Learning and Statistics

Statistics is a branch of mathematics that has a solid theoretical foundation regarding statistical learning. Like machine learning (ML), it can learn from data. But the difference between statistics and ML is thatstatistical methods look for regularity in data called patterns. Initially, statistics sets a hypothesis and performs experiments to verify and validate the hypothesis in order to find relationships among data.

Statistics requires knowledge of the statistical procedures and the guidance of a good statistician. It is mathematics intensive and models are often complicated equations and involve many assumptions. Statistical methods are developed in relation to the data being analyzed. In addition, statistical methods are coherent and rigorous. It has strong theoretical foundations and interpretations that require a strong statistical knowledge.

Machine learning, comparatively, has less assumptions and requires less statistical knowledge. But, it often requires interaction with various tools to automate the process of learning.

Nevertheless, there is a school of thought that machine learning is just the latest version of 'old Statistics' and hence this relationship should be recognized.

1.4 TYPES OF MACHINE LEARNING

What does the word 'learn' mean? Learning, like adaptation, occurs as the result of interaction of the program with its environment. It can be compared with the interaction between a teacher and a student. There are four types of machine learning as shown in Figure 1.5.

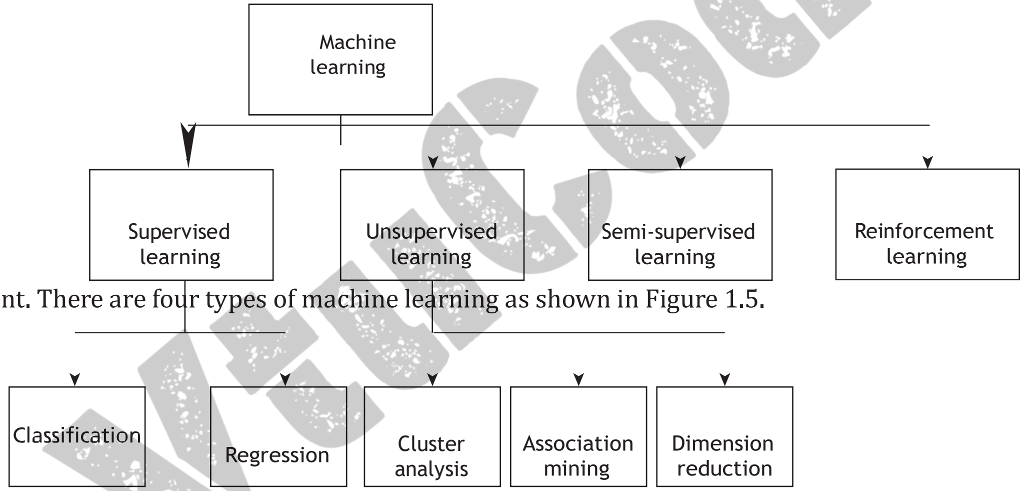


Figure 1.5: Types of Machine Learning

Before discussing the types of learning, it is necessary to discuss about data.

Labelled and Unlabeled Data Data is a raw fact. Normally, data is represented in the formof a table. Data also can be referred to as a data point, sample, or an example. Each row of the table represents a data point. Features are attributes or characteristics of an object. Normally, the columns of the table are attributes. Out of all attributes, one attribute is important and is called a label. Label is the feature that we aim to predict. Thus, there are two types of data – labelled and unlabeled.

Labelled Data To illustrate labelled data, let us take one example dataset called Iris flower datasetor Fisher's Iris dataset. The dataset has 50 samples of Iris – with four attributes, length and width of sepals and petals. The target variable is called class. There are three classes – Iris setosa, Iris virginica, and Iris versicolor.

The partial data of Iris dataset is shown in Table 1.1.

Table 1.1: Iris Flower Dataset

S.No.	Length of Petal	Width of Petal	Length of Sepal	Width of Sepal	Class
1.	5.5	4.2	1.4	0.2	Setosa
2.	7	3.2	4.7	1.4	Versicolor
3.	7.3	2.9	6.3	1.8	Virginica

A dataset need not be always numbers. It can be images or video frames. Deep neural networkscan handle images with labels. In the following Figure 1.6, the deep neural network takes images ofdogs and cats with labels for classification.

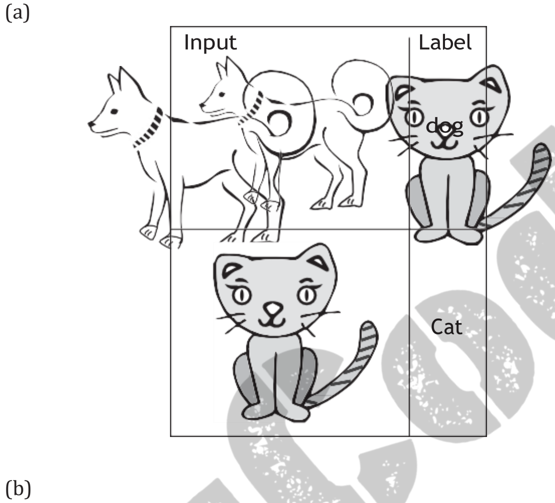


Figure 1.6: (a) Labelled Dataset (b) Unlabeled Dataset

In unlabeled data, there are no labels in the dataset.

1.4.1 Supervised Learning

Supervised algorithms use labelled dataset. As the name suggests, there is a supervisor or teacher component in supervised learning. A supervisor provides labelled data so that the model is constructed and generates test data.

In supervised learning algorithms, learning takes place in two stages. In layman terms, during thefirst stage, the teacher communicates the information to the student that the student is supposed tomaster. The student receives the information and understands it. During this stage, the teacher has noknowledge of whether the information is grasped by the student.

This leads to the second stage of learning. The teacher then asks the student a set of questionsto find out how much information has been grasped by the student. Based on these questions, the student is tested, and the teacher informs the student about his assessment. This kind of learningis typically called supervised learning.

Supervised learning has two methods:

1. Classification
2. Regression

Classification

Classification is a supervised learning method. The input attributes of the classification algorithmsare called independent variables. The target attribute is called label or dependent variable. The relationship between the input and target variable is represented in the form of a structure which is called a classification model. So, the focus of classification is to predict the 'label' that is in a discrete form (a value from the set of finite values). An example is shown in Figure 1.7 where a classification algorithm takes a set of labelled data images such as dogs and cats to construct a model that can later be used to classify an unknown test image data.

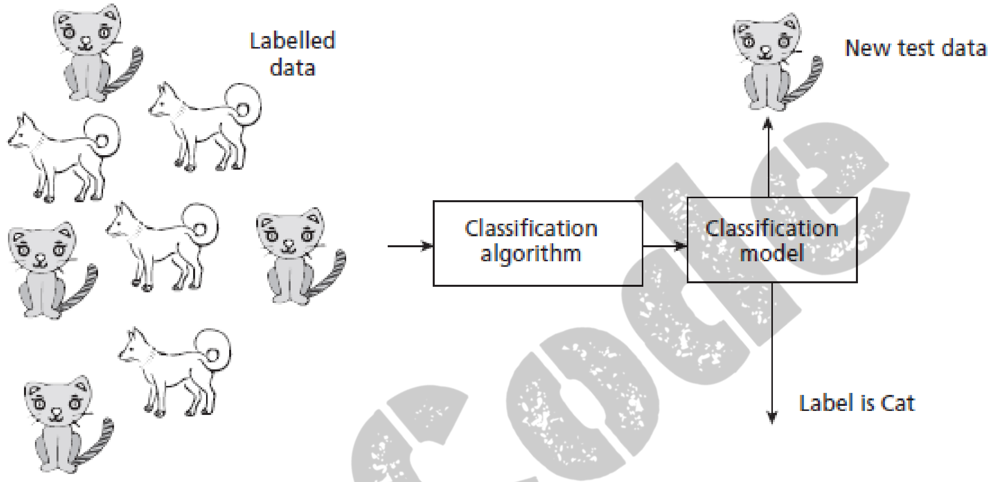


Figure 1.7: An Example Classification System

In classification, learning takes place in two stages. During the first stage, called training stage,the learning algorithm takes a labelled dataset and starts learning. After the training set, samples are processed and the model is generated. In the second stage, the constructed model is tested withtest or unknown sample and assigned a label. This is the classification process.

This is illustrated in the above Figure 1.7. Initially, the classification learning algorithm learnswith the collection of labelled data and constructs the model. Then, a test case is selected, and the model assigns a label.

Similarly, in the case of Iris dataset, if the test is given as (6.3, 2.9, 5.6, 1.8, ?), the classification will generate the label for this. This is called classification. One of the examples of classification is –Image recognition, which includes classification of diseases like cancer, classification of plants, etc.

The classification models can be categorized based on the implementation technology like decision trees, probabilistic methods, distance measures, and soft computing methods. Classificationmodels can also be classified as generative models and discriminative models. Generative modelsdeal with the process of data generation and its distribution. Probabilistic models are examples of generative models. Discriminative models do not care about the generation of data. Instead, they simply concentrate on classifying the given data.

Some of the key algorithms of classification are:

- Decision Tree
- Random Forest
- Support Vector Machines
- Naïve Bayes
- Artificial Neural Network and Deep Learning networks like CNN

Regression Models

Regression models, unlike classification algorithms, predict continuous variables like price. In other words, it is a number. A fitted regression model is shown in Figure 1.8 for a dataset that represent weeks input *x* and product sales *y*.

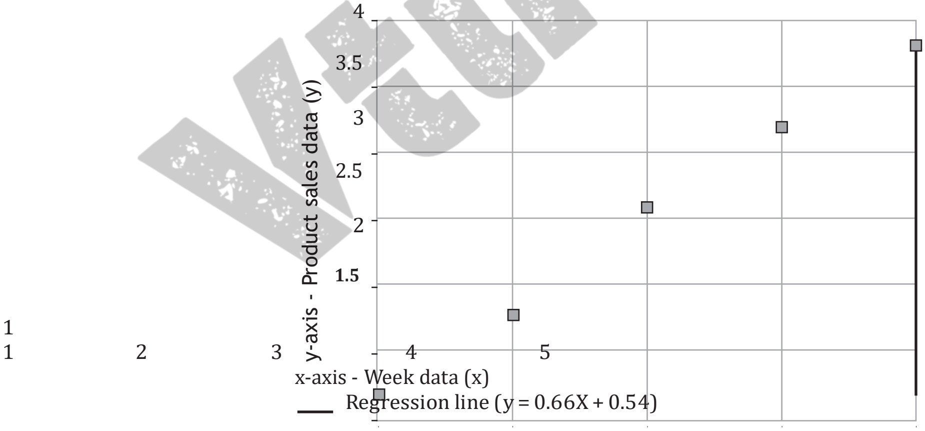


Figure 1.8: A Regression Model of the Form *y* = *ax* + *b*

The regression model takes input *x* and generates a model in the form of a fitted line of the form *y* = *f*(*x*). Here, *x* is the independent variable that may be one or more attributes and *y* is the dependent variable. In Figure 1.8, linear regression takes the training set and tries to fit it with a line – product sales = 0.66 ⍰ Week + 0.54. Here, 0.66 and 0.54 are all regression coefficients that arelearnt from data. The advantage of this model is that prediction for product sales (*y*) can be made for unknown week data (x). For example, the prediction for unknown eighth week can be made bysubstituting x as 8 in that regression formula to get y.

One of the most important regression algorithms is linear regression that is explained in the next section.

Both regression and classification models are supervised algorithms. Both have a supervisor andthe concepts of training and testing are applicable to both. What is the difference between classificationand regression models? The main difference is that regression models predict continuous variablessuch as product price, while classification concentrates on assigning labels such as class.

1.4.2 Unsupervised Learning

The second kind of learning is by self-instruction. As the name suggests, there are no supervisor or teacher components. In the absence of a supervisor or teacher, self-instruction is the most commonkind of learning process. This process of self-instruction is based on the concept of trial and error.

Here, the program is supplied with objects, but no labels are defined. The algorithm itself observes the examples and recognizes patterns based on the principles of grouping. Grouping is done in ways that similar objects form the same group.

Cluster analysis and Dimensional reduction algorithms are examples of unsupervised algorithms.

Cluster Analysis

Cluster analysis is an example of unsupervised learning. It aims to group objects into disjoint clusters or groups. Cluster analysis clusters objects based on its attributes. All the data objectsof the partitions are similar in some aspect and vary from the data objects in the other partitions significantly.

Some of the examples of clustering processes are — segmentation of a region of interest in an image, detection of abnormal growth in a medical image, and determining clusters of signatures in a gene database.

An example of clustering scheme is shown in Figure 1.9 where the clustering algorithm takes a set of dogs and cats images and groups it as two clusters-dogs and cats. It can be observed that the samples belonging to a cluster are similar and samples are different radically across clusters.

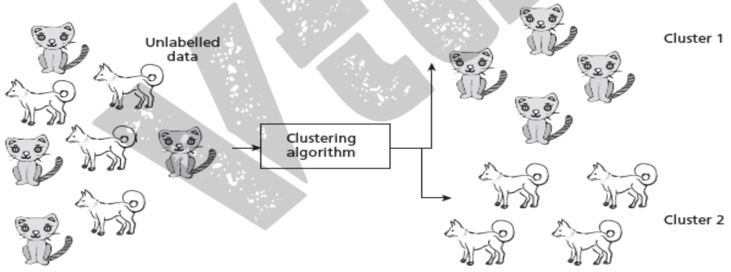


Figure 1.9: An Example Clustering Scheme

Some of the key clustering algorithms are:

- k-means algorithm
- Hierarchical algorithms

Dimensionality Reduction

Dimensionality reduction algorithms are examples of unsupervised algorithms. It takes a higher dimension data as input and outputs the data in lower dimension by taking advantage of the varianceof the data. It is a task of reducing the dataset with few features without losing the generality.

The differences between supervised and unsupervised learning are listed in the following Table 1.2.

Table 1.2: Differences between Supervised and Unsupervised Learning

S.No.	Supervised Learning	Unsupervised Learning
1.	There is a supervisor component	No supervisor component
2.	Uses Labelled data	Uses Unlabelled data
3.	Assigns categories or labels	Performs grouping process such that similar objectswill be in one cluster

1.4.3 Semi-supervised Learning

There are circumstances where the dataset has a huge collection of unlabelled data and some labelled data. Labelling is a costly process and difficult to perform by the humans. Semi-supervisedalgorithms use unlabelled data by assigning a pseudo-label. Then, the labelled and pseudo-labelled dataset can be combined.

1.4.4 Reinforcement Learning

Reinforcement learning mimics human beings. Like human beings use ears and eyes to perceive theworld and take actions, reinforcement learning allows the agent to interact with the environment to get rewards. The agent can be human, animal, robot, or any independent program. The rewardsenable the agent to gain experience. The agent aims to maximize the reward.

The reward can be positive or negative (Punishment). When the rewards are more, the behaviorgets reinforced and learning becomes possible.

Consider the following example of a Grid game as shown in Figure 1.10.

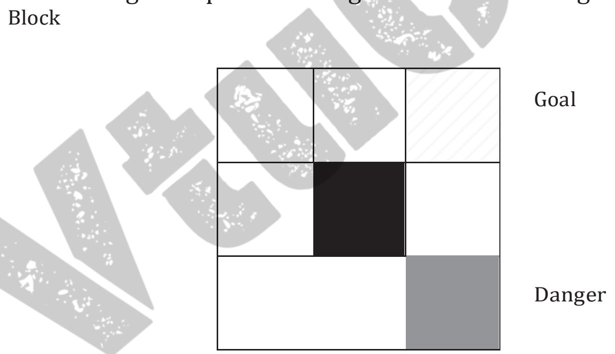


Figure 1.10: A Grid game

In this grid game, the gray tile indicates the danger, black is a block, and the tile with diagonallines is the goal. The aim is to start, say from bottom-left grid, using the actions left, right, top andbottom to reach the goal state.

To solve this sort of problem, there is no data. The agent interacts with the environment toget experience. In the above case, the agent tries to create a model by simulating many paths and finding rewarding paths. This experience helps in constructing a model.

It can be said in summary, compared to supervised learning, there is no supervisor orlabelled dataset. Many sequential decisions need to be taken to reach the final decision. Therefore, reinforcement algorithms are reward-based, goal-oriented algorithms.

1.5 CHALLENGES OF MACHINE LEARNING

What are the challenges of machine learning? Let us discuss about them now.

Problems that can be Dealt with Machine Learning

Computers are better than humans in performing tasks like computation. For example, while calculatingthe square root of large numbers, an average human may blink but computers can display the result inseconds. Computers can play games like chess, GO, and even beat professional players of that game.

However, humans are better than computers in many aspects like recognition. But, deep learning systems challenge human beings in this aspect as well. Machines can recognize human faces in a second. Still, there are tasks where humans are better as machine learning systems still require quality data for model construction. The quality of a learning system depends on the quality of data. This is a challenge. Some of the challenges are listed below:

1. Problems – Machine learning can deal with the 'well-posed' problems where specificationsare complete and available. Computers cannot solve 'ill-posed' problems.

   Consider one simple example (shown in Table 1.3):

   Table 1.3: An Example

Input (x_1, x_2)	Output (y)
1, 1	1
2, 1	2
3, 1	3
4, 1	4
5, 1	5

   Can a model for this test data be multiplication? That is, $y ⍰ x_1 ⍰ x_2$. Well! It is true! But, this is equally true that y may be $y ⍰ x_1 ⍰ x_2$, or $y ⍰ x_1^{x_2}$. So, there are three functions that fit the data.

   This means that the problem is ill-posed. To solve this problem, one needs more example to check the model. Puzzles and games that do not have sufficient specification may become anill-posed problem and scientific computation has many ill-posed problems.

2. Huge data – This is a primary requirement of machine learning. Availability of a quality data is a challenge. A quality data means it should be large and should not have data problems such as missing data or incorrect data.
3. High computation power – With the availability of Big Data, the computational resource requirement has also increased. Systems with *Graphics Processing Unit* (GPU) or even *Tensor Processing Unit* **(**TPU) are required to execute machine learning algorithms. Also, machine learning tasks have become complex and hence time complexity has increased, and that can be solved only with high computing power.
4. Complexity of the algorithms – The selection of algorithms, describing the algorithms, application of algorithms to solve machine learning task, and comparison of algorithms have become necessary for machine learning or data scientists now. Algorithms have become a big topic of discussion and it is a challenge for machine learning professionals todesign, select, and evaluate optimal algorithms.
5. Bias/Variance – Variance is the error of the model. This leads to a problem called bias/ variance tradeoff. A model that fits the training data correctly but fails for test data, in general lacks generalization, is called overfitting. The reverse problem is called underfitting where the model fails for training data but has good generalization. Overfitting and underfitting are great challenges for machine learning algorithms.

1.6 MACHINE LEARNING PROCESS

The emerging process model for the data mining solutions for business organizations is CRISP-DM.Since machine learning is like data mining, except for the aim, this process can be used for machinelearning. CRISP-DM stands for Cross Industry Standard Process – Data Mining. This process involves six steps. The steps are listed below in Figure 1.11.

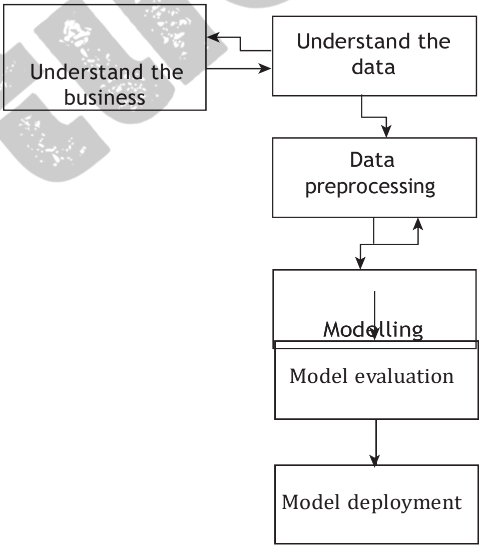


Figure 1.11: A Machine Learning/Data Mining Process

1. Understanding the business – This step involves understanding the objectives and requirements of the business organization. Generally, a single data mining algorithm is enough for giving the solution. This step also involves the formulation of the problem statement for the data mining process.
2. Understanding the data – It involves the steps like data collection, study of the charac teristics of the data, formulation of hypothesis, and matching of patterns to the selected hypothesis.
3. Preparation of data – This step involves producing the final dataset by cleaning the raw data and preparation of data for the data mining process. The missing values may cause problems during both training and testing phases. Missing data forces classifiers to produceinaccurate results. This is a perennial problem for the classification models. Hence, suitablestrategies should be adopted to handle the missing data.
4. Modelling – This step plays a role in the application of data mining algorithm for the datato obtain a model or pattern.
5. Evaluate – This step involves the evaluation of the data mining results using statistical analysis and visualization methods. The performance of the classifier is determined by evaluating the accuracy of the classifier. The process of classification is a fuzzy issue.For example, classification of emails requires extensive domain knowledge and requires domain experts. Hence, performance of the classifier is very crucial.
6. Deployment – This step involves the deployment of results of the data mining algorithm to improve the existing process or for a new situation.

1.7 MACHINE LEARNING APPLICATIONS

Machine Learning technologies are used widely now in different domains. Machine learning applications are everywhere! One encounters many machine learning applications in the day-to-day life. Some applications are listed below:

1. Sentiment analysis – This is an application of natural language processing (NLP) where the words of documents are converted to sentiments like happy, sad, and angry which arecaptured by emoticons effectively. For movie reviews or product reviews, five stars or onestar are automatically attached using sentiment analysis programs.
2. Recommendation systems – These are systems that make personalized purchases possible.For example, Amazon recommends users to find related books or books bought by peoplewho have the same taste like you, and Netflix suggests shows or related movies of your taste. The recommendation systems are based on machine learning.
3. Voice assistants – Products like Amazon Alexa, Microsoft Cortana, Apple Siri, and Google Assistant are all examples of voice assistants. They take speech commands and perform tasks. These chatbots are the result of machine learning technologies.
4. Technologies like Google Maps and those used by Uber are all examples of machine learning which offer to locate and navigate shortest paths to reduce time.

The machine learning applications are enormous. The following Table 1.4 summarizes some ofthe machine learning applications.

Table 1.4: Applications' Survey Table

S.No.	Problem Domain	Applications
1.	Business	Predicting the bankruptcy of a business firm
2.	Banking	Prediction of bank loan defaulters and detecting credit card frauds
3.	Image Processing	Image search engines, object identification, image classification, and generating synthetic images
4.	Audio/Voice	Chatbots like Alexa, Microsoft Cortana. Developing chatbots forcustomer support, speech to text, and text to voice
5.	Telecommunication	Trend analysis and identification of bogus calls, fraudulent calls and its callers, churn analysis
6.	Marketing	Retail sales analysis, market basket analysis, product performance analysis, market segmentation analysis, and study of travel patterns of customers for marketing tours
7.	Games	Game programs for Chess, GO, and Atari video games
8.	Natural Language Translation	Google Translate, Text summarization, and sentiment analysis
9.	Web Analysis and Services	Identification of access patterns, detection of e-mail spams, viruses, personalized web services, search engines like Google, detection of promotion of user websites, and finding loyalty of users after web page layout modification
10.	Medicine	Prediction of diseases, given disease symptoms as cancer or diabetes. Prediction of effectiveness of the treatment using patient history and Chatbots to interact with patients like IBM Watson uses machinelearning technologies.
11.	Multimedia and Security	Face recognition/identification, biometric projects like identificationof a person from a large image or video database, and applications involving multimedia retrieval
12.	Scientific Domain	Discovery of new galaxies, identification of groups of houses basedon house type/geographical location, identification of earthquake epicenters, and identification of similar land use

Key Terms:

- **Machine Learning** – A branch of AI that concerns about machines to learn automatically withoutbeing explicitly programmed.
- **Data** – A raw fact.
- **Model** – An explicit description of patterns in a data.
- **Experience** – A collection of knowledge and heuristics in humans and historical training data in case of machines.
- **Predictive Modelling** – A technique of developing models and making a prediction of unseen data.
- **Deep Learning** – A branch of machine learning that deals with constructing models using neural networks.
- **Data Science** – A field of study that encompasses capturing of data to its analysis covering all stagesof data management.
- **Data Analytics** – A field of study that deals with analysis of data.

- **Big Data** – A study of data that has characteristics of volume, variety, and velocity.
- **Statistics** – A branch of mathematics that deals with learning from data using statistical methods.
- **Hypothesis** – An initial assumption of an experiment.
- **Learning** – Adapting to the environment that happens because of interaction of an agent with the environment.
- **Label** – A target attribute.
- **Labelled Data** – A data that is associated with a label.
- **Unlabelled Data** – A data without labels.
- **Supervised Learning** – A type of machine learning that uses labelled data and learns with the help of a supervisor or teacher component.
- **Classification Program** – A supervisory learning method that takes an unknown input and assigns a label for it. In simple words, finds the category of class of the input attributes.
- **Regression Analysis** – A supervisory method that predicts the continuous variables based on the input variables.
- **Unsupervised Learning** – A type of machine leaning that uses unlabelled data and groups the attributes to clusters using a trial and error approach.
- **Cluster Analysis** – A type of unsupervised approach that groups the objects based on attributesso that similar objects or data points form a cluster.
- **Semi-supervised Learning** – A type of machine learning that uses limited labelled and largeunlabelled data. It first labels unlabelled data using labelled data and combines it for learning purposes.
- **Reinforcement Learning** – A type of machine learning that uses agents and environment interactionfor creating labelled data for learning.
- **Well-posed Problem** – A problem that has well-defined specifications. Otherwise, the problem is called ill-posed.
- **Bias/Variance** – The inability of the machine learning algorithm to predict correctly due to lackof generalization is called bias. Variance is the error of the model for training data. This leads to problems called overfitting and underfitting.
- **Model Deployment** – A method of deploying machine learning algorithms to improve the existing business processes for a new situation.

2.1 WHAT IS DATA?

All facts are data. In computer systems, bits encode facts present in numbers, text, images, audio, and video. Data can be directly human interpretable (such as numbers or texts) or diffused data such as images or video that can be interpreted only by a computer.

Data is available in different data sources like flat files, databases, or data warehouses. It can either be an operational data or a non-operational data. Operational data is the one that is encountered in normal business procedures and processes. For example, daily sales data is operational data, on the other hand, non-operational data is the kind of data that is used for decision making.

Data by itself is meaningless. It has to be processed to generate any information. A string of bytes is meaningless. Only when a label is attached like height of students of a class, the data becomes meaningful. Processed data is called information that includes patterns, associations, or relationships among data. For example, sales data can be analyzed to extract information like which product was sold larger in the last quarter of the year.

Elements of Big Data

Data whose volume is less and can be stored and processed by a small-scale computer is called 'small data'. These data are collected from several sources, and integrated and processed by a small-scale computer. Big data, on the other hand, is a larger data whose volume is much larger than 'small data' and is characterized as follows:

1. Volume – Since there is a reduction in the cost of storing devices, there has been a tremendous growth of data. Small traditional data is measured in terms of gigabytes (GB) and terabytes (TB), but Big Data is measured in terms of petabytes (PB) and exabytes (EB). One exabyte is 1 million terabytes.
2. Velocity – The fast arrival speed of data and its increase in data volume is noted as velocity. The availability of IoT devices and Internet power ensures that the data is arriving at a faster rate. Velocity helps to understand the relative growth of big data and its accessibility by users, systems and applications.
3. Variety – The variety of Big Data includes:
    - Form – There are many forms of data. Data types range from text, graph, audio, video, to maps. There can be composite data too, where one media can have many other sources of data, for example, a video can have an audio song.
    - Function – These are data from various sources like human conversations, transaction records, and old archive data.
    - Source of data – This is the third aspect of variety. There are many sources of data. Broadly, the data source can be classified as open/public data, social media data and multimodal data.

Some of the other forms of Vs that are often quoted in the literature as characteristics of Big data are:

4. Veracity of data – Veracity of data deals with aspects like conformity to the facts, truthfulness, believablity, and confidence in data. There may be many sources of error such as technical errors, typographical errors, and human errors. So, veracity is one of the most important aspects of data.
5. Validity – Validity is the accuracy of the data for taking decisions or for any other goals that are needed by the given problem.
6. Value – Value is the characteristic of big data that indicates the value of the information that is extracted from the data and its influence on the decisions that are taken based on it.

Thus, these 6 Vs are helpful to characterize the big data. The data quality of the numeric attributes is determined by factors like precision, bias, and accuracy.

- Precision is defined as the closeness of repeated measurements. Often, standard deviation is used to measure the precision.
- Bias is a systematic result due to erroneous assumptions of the algorithms or procedures.
- Accuracy is the degree of measurement of errors that refers to the closeness of measurements to the true value of the quantity. Normally, the significant digits used to store and manipulate indicate the accuracy of the measurement.

2.1.1 Types of Data

In Big Data, there are three kinds of data. They are structured data, unstructured data, and semi-structured data.

Structured Data

In structured data, data is stored in an organized manner such as a database where it is available in the form of a table. The data can also be retrieved in an organized manner using tools like SQL. The structured data frequently encountered in machine learning are listed below:

Record Data A dataset is a collection of measurements taken from a process. We have a collection of objects in a dataset and each object has a set of measurements. The measurements can be arranged in the form of a matrix. Rows in the matrix represent an object and can be called as entities, cases, or records. The columns of the dataset are called attributes, features, or fields. The table is filled with observed data. Also, it is better to note the general jargons that are associated with the dataset. Label is the term that is used to describe the individual observations.

Data Matrix It is a variation of the record type because it consists of numeric attributes. The standard matrix operations can be applied on these data. The data is thought of as points or vectors in the multidimensional space where every attribute is a dimension describing the object.

Graph Data It involves the relationships among objects. For example, a web page can refer to another web page. This can be modeled as a graph. The modes are web pages and the hyperlink is an edge that connects the nodes.

Ordered Data Ordered data objects involve attributes that have an implicit order among them.
The examples of ordered data are:

- Temporal data – It is the data whose attributes are associated with time. For example, the customer purchasing patterns during festival time is sequential data. Time series data is a special type of sequence data where the data is a series of measurements over time.

- Sequence data – It is like sequential data but does not have time stamps. This data involves the sequence of words or letters. For example, DNA data is a sequence of four characters – A T G C.

- Spatial data – It has attributes such as positions or areas. For example, maps are spatial data where the points are related by location.

Unstructured Data

Unstructured data includes video, image, and audio. It also includes textual documents, programs, and blog data. It is estimated that 80% of the data are unstructured data.

Semi-Structured Data

Semi-structured data are partially structured and partially unstructured. These include data like XML/JSON data, RSS feeds, and hierarchical data.

2.1.2 Data Storage and Representation

Once the dataset is assembled, it must be stored in a structure that is suitable for data analysis. The goal of data storage management is to make data available for analysis. There are different approaches to organize and manage data in storage files and systems from flat file to data warehouses. Some of them are listed below:

Flat Files These are the simplest and most commonly available data source. It is also the cheapest way of organizing the data. These flat files are the files where data is stored in plain ASCII or EBCDIC format. Minor changes of data in flat files affect the results of the data mining algorithms.
Hence, flat file is suitable only for storing small dataset and not desirable if the dataset becomes larger.
Some of the popular spreadsheet formats are listed below:

- CSV files – CSV stands for comma-separated value files where the values are separated by commas. These are used by spreadsheet and database applications. The first row may have attributes and the rest of the rows represent the data.
- TSV files – TSV stands for Tab separated values files where values are separated by Tab. Both CSV and TSV files are generic in nature and can be shared. There are many tools like Google Sheets and Microsoft Excel to process these files.

Database System It normally consists of database files and a database management system (DBMS). Database files contain original data and metadata. DBMS aims to manage data and improve operator performance by including various tools like database administrator, query processing, and transaction manager. A relational database consists of sets of tables. The tables have rows and columns. The columns represent the attributes and rows represent tuples. A tuple corresponds to either an object or a relationship between objects. A user can access and manipulate the data in the database using SQL.

Different types of databases are listed below:

1 A transactional database is a collection of transactional records. Each record is a transaction. A transaction may have a time stamp, identifier and a set of items, which may have links to other tables. Normally, transaction databases are created for performing associational analysis that indicates the correlation among the items.

2. Time-series database stores time related information like log files where data is associated with a time stamp. This data represents the sequences of data, which represent values or events obtained over a period (for example, hourly, weekly or yearly) or repeated time span. Observing sales of product continuously may yield a time-series data.

3. Spatial databases contain spatial information in a raster or vector format. Raster formats are either bitmaps or pixel maps. For example, images can be stored as a raster data. On the other hand, the vector format can be used to store maps as maps use basic geometric primitives like points, lines, polygons and so forth.

World Wide Web (WWW) It provides a diverse, worldwide online information source.
The objective of data mining algorithms is to mine interesting patterns of information present in WWW.
XML (eXtensible Markup Language) It is both human and machine interpretable data format that can be used to represent data that needs to be shared across the platforms.
Data Stream It is dynamic data, which flows in and out of the observing environment. Typical characteristics of data stream are huge volume of data, dynamic, fixed order movement, and real-time constraints.
RSS (Really Simple Syndication) It is a format for sharing instant feeds across services.
JSON (JavaScript Object Notation) It is another useful data interchange format that is often used for many machine learning algorithms.

2.2 BIG DATA ANALYTICS AND TYPES OF ANALYTICS

The primary aim of data analysis is to assist business organizations to take decisions. For example, a business organization may want to know which is the fastest selling product, in order for them to market activities. Data analysis is an activity that takes the data and generates useful information and insights for assisting the organizations.

Data analysis and data analytics are terms that are used interchangeably to refer to the same concept. However, there is a subtle difference. Data analytics is a general term and data analysis is a part of it. Data analytics refers to the process of data collection, preprocessing and analysis. It deals with the complete cycle of data management. Data analysis is just analysis and is a part of data analytics. It takes historical data and does the analysis.

Data analytics, instead, concentrates more on future and helps in prediction.

There are four types of data analytics:

1. Descriptive analytics
2. Diagnostic analytics
3. Predictive analytics
4. Prescriptive analytics

Descriptive Analytics It is about describing the main features of the data. After data collection is done, descriptive analytics deals with the collected data and quantifies it. It is often stated that analytics is essentially statistics. There are two aspects of statistics – Descriptive and Inference. Descriptive analytics only focuses on the description part of the data and not the inference part.

Diagnostic Analytics It deals with the question – 'Why?'. This is also known as causal analysis, as it aims to find out the cause and effect of the events. For example, if a product is not selling, diagnostic analytics aims to find out the reason. There may be multiple reasons and associated effects are analyzed as part of it.

Predictive Analytics It deals with the future. It deals with the question – 'What will happen in future given this data?'. This involves the application of algorithms to identify the patterns to predict the future. The entire course of machine learning is mostly about predictive analytics and forms the core of this book.

Prescriptive Analytics It is about the finding the best course of action for the business organizations. Prescriptive analytics goes beyond prediction and helps in decision making by giving a set of actions. It helps the organizations to plan better for the future and to mitigate the risks that are involved.

2.3 BIG DATA ANALYSIS FRAMEWORK

For performing data analytics, many frameworks are proposed. All proposed analytics frameworks have some common factors. Big data framework is a layered architecture. Such an architecture has many advantages such as genericness. A 4-layer architecture has the following layers:

1. Date connection layer
2. Data management layer
3. Data analytics later
4. Presentation layer

Data Connection Layer It has data ingestion mechanisms and data connectors. Data ingestion means taking raw data and importing it into appropriate data structures. It performs the tasks of **ETL process. By ETL, it means extract, transform and load operations.**

Data Management Layer It performs preprocessing of data. The purpose of this layer is to allow parallel execution of queries, and read, write and data management tasks. There may be many schemes that can be implemented by this layer such as data-in-place, where the data is not moved at all, or constructing data repositories such as data warehouses and pull data on-demand mechanisms.

Data Analytic Layer It has many functionalities such as statistical tests, machine learning algorithms to understand, and construction of machine learning models. This layer implements many model validation mechanisms too. The processing is done as shown in Box 2.1.

Presentation Layer It has mechanisms such as dashboards, and applications that display the

results of analytical engines and machine learning algorithms.
Thus, the Big Data processing cycle involves data management that consists of the following steps.

1. Data collection
2. Data preprocessing
3. Applications of machine learning algorithm
4. Interpretation of results and visualization of machine learning algorithm

This is an iterative process and is carried out on a permanent basis to ensure that data is suitable for data mining.
Application and interpretation of machine learning algorithms constitute the basis for the rest of the book. So, primarily, data collection and data preprocessing are covered as part of this chapter.

2.3.1 Data Collection

The first task of gathering datasets are the collection of data. It is often estimated that most of the time is spent for collection of good quality data. A good quality data yields a better result. It is often difficult to characterize a 'Good data'. 'Good data' is one that has the following properties:

1. Timeliness – The data should be relevant and not stale or obsolete data.
2. Relevancy – The data should be relevant and ready for the machine learning or data mining algorithms. All the necessary information should be available and there should be no bias in the data.
3. Knowledge about the data – The data should be understandable and interpretable, and should be self-sufficient for the required application as desired by the domain knowledge engineer.

Broadly, the data source can be classified as open/public data, social media data and multimodal data.

1. Open or public data source – It is a data source that does not have any stringent copyright rules or restrictions. Its data can be primarily used for many purposes. Government census data are good examples of open data:
   - Digital libraries that have huge amount of text data as well as document images
   - Scientific domains with a huge collection of experimental data like genomic data and biological data
   - Healthcare systems that use extensive databases like patient databases, health insurance data, doctors' information, and bioinformatics information
2. Social media – It is the data that is generated by various social media platforms like Twitter, Facebook, YouTube, and Instagram. An enormous amount of data is generated by these platforms.
3. Multimodal data – It includes data that involves many modes such as text, video, audio and mixed types. Some of them are listed below:
   - Image archives contain larger image databases along with numeric and text data
   - The World Wide Web (WWW) has huge amount of data that is distributed on the Internet.

These data are heterogeneous in nature.

2.3.2 Data Preprocessing

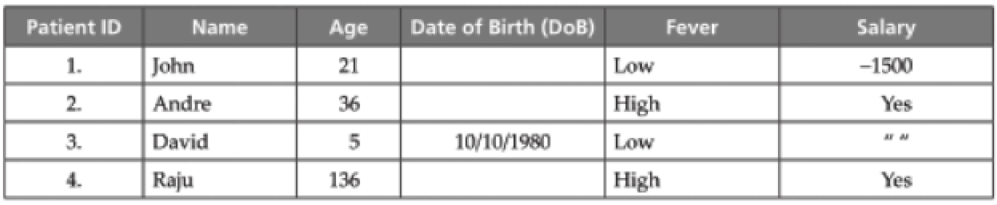
In real world, the available data is 'dirty'. By this word 'dirty', it means:

- Incomplete data
- Outlier data
- Data with inconsistent values
- Inaccurate data
- Data with missing values
- Duplicate data

Data preprocessing improves the quality of the data mining techniques. The raw data must be preprocessed to give accurate results. The process of detection and removal of errors in data is called data cleaning. Data wrangling means making the data processable for machine learning algorithms. Some of the data errors include human errors such as typographical errors or incorrect

measurement and structural errors like improper data formats. Data errors can also arise from omission and duplication of attributes. Noise is a random component and involves distortion of a value or introduction of spurious objects. Often, the noise is used if the data is a spatial or temporal component. Certain deterministic distortions in the form of a streak are known as artifacts.
Consider, for example, the following patient Table 2.1. The ‘bad’ or ‘dirty’ data can be observed in this table.

Table 2.1: Illustration of ‘Bad’ Data



Patient ID	Name	Age	Date of Birth (DoB)	Fever	Salary
1.	John	21		Low	−1500
2.	Andre	36		High	Yes
3.	David	5	10/10/1980	Low	“ ”
4.	Raju	136		High	Yes

It can be observed that data like Salary = ’ ’ is incomplete data. The DoB of patients, John, Andre, and Raju, is the missing data. The age of David is recorded as ‘5’ but his DoB indicates it is 10/10/1980. This is called inconsistent data.

Inconsistent data occurs due to problems in conversions, inconsistent formats, and difference in units. Salary for John is -1500. It cannot be less than ‘0’. It is an instance of noisy data. Outliers are data that exhibit the characteristics that are different from other data and have very unusual values. The age of Raju cannot be 136. It might be a typographical error. It is often required to distinguish between noise and outlier data.
Outliers may be legitimate data and sometimes are of interest to the data mining algorithms. These errors often come during data collection stage. These must be removed so that machine learning algorithms yield better results as the quality of results is determined by the quality of input data. This removal process is called data cleaning.

Missing Data Analysis
The primary data cleaning process is missing data analysis. Data cleaning routines attempt to fill up the missing values, smoothen the noise while identifying the outliers and correct the inconsistencies of the data. This enables data mining to avoid overfitting of the models.
The procedures that are given below can solve the problem of missing data:
1. Ignore the tuple – A tuple with missing data, especially the class label, is ignored. This method is not effective when the percentage of the missing values increases.
2. Fill in the values manually – Here, the domain expert can analyse the data tables and carry out the analysis and fill in the values manually. But, this is time consuming and may not be feasible for larger sets.
3. A global constant can be used to fill in the missing attributes. The missing values may be ’Unknown’ or be ’Infinity’. But, some data mining results may give spurious results by analysing these labels.
4. The attribute value may be filled by the attribute value. Say, the average income can replace a missing value.
5. Use the attribute mean for all samples belonging to the same class. Here, the average value replaces the missing values of all tuples that fall in this group.
6. Use the most possible value to fill in the missing value. The most probable value can be obtained from other methods like classification and decision tree prediction.
Some of these methods introduce bias in the data. The filled value may not be correct and could be just an estimated value. Hence, the difference between the estimated and the original value is called an error or bias.

Removal of Noisy or Outlier Data

Noise is a random error or variance in a measured value. It can be removed by using binning, which is a method where the given data values are sorted and distributed into equal frequency bins. The bins are also called as buckets. The binning method then uses the neighbor values to smooth the noisy data.

Some of the techniques commonly used are 'smoothing by means' where the mean of the bin removes the values of the bins, 'smoothing by bin medians' where the bin median replaces the bin values, and 'smoothing by bin boundaries' where the bin value is replaced by the closest bin boundary. The maximum and minimum values are called bin boundaries. Binning methods may be used as a discretization technique. Example 2.1 illustrates this principle.

Example 2.1: Consider the following set: S = {12, 14, 19, 22, 24, 26, 28, 31, 34}. Apply various binning techniques and show the result.

Solution: By equal-frequency bin method, the data should be distributed across bins. Let us assume the bins of size 3, then the above data is distributed across the bins as shown below:

Bin 1 : 12 , 14, 19
Bin 2 : 22, 24, 26
Bin 3 : 28, 31, 32

By smoothing bins method, the bins are replaced by the bin means. This method results in:

Bin 1 : 15, 15, 15
Bin 2 : 24, 24, 24
Bin 3 : 30.3, 30.3, 30.3

Using smoothing by bin boundaries method, the bins' values would be like:

Bin 1 : 12, 12, 19
Bin 2 : 22, 22, 26
Bin 3 : 28, 32, 32

As per the method, the minimum and maximum values of the bin are determined, and it serves as bin boundary and does not change. Rest of the values are transformed to the nearest value. It can be observed in Bin 1, the middle value 14 is compared with the boundary values 12 and 19 and changed to the closest value, that is 12. This process is repeated for all bins.

Data Integration and Data Transformations

Data integration involves routines that merge data from multiple sources into a single data source. So, this may lead to redundant data. The main goal of data integration is to detect and remove redundancies that arise from integration. Data transformation routines perform operations like normalization to improve the performance of the data mining algorithms. It is necessary to transform data so that it can be processed. This can be considered as a preliminary stage of data conditioning. Normalization is one such technique. In normalization, the attribute values are scaled to fit in a range (say 0-1) to improve the performance of the data mining algorithm. Often, in neural networks, these techniques are used. Some of the normalization procedures used are:

1. Min-Max
2. z-Score

Min-Max Procedure It is a normalization technique where each variable V is normalized by its difference with the minimum value divided by the range to a new range, say 0–1. Often, neural networks require this kind of normalization. The formula to implement this normalization is given as:

$$\textit{min-max} = \frac{V - \textit{min}}{\textit{max} - \textit{min}} \times (\textit{new max} - \textit{new min}) + \textit{new min} \qquad (2.1)$$

Here max-min is the range. Min and max are the minimum and maximum of the given data, new max and new min are the minimum and maximum of the target range, say 0 and 1.

Example 2.2: Consider the set: V = {88, 90, 92, 94}. Apply Min-Max procedure and map the marks

to a new range 0–1.
Solution: The minimum of the list V is 88 and maximum is 94. The new min and new max are 0 and 1, respectively. The mapping can be done using Eq. (2.1) as:

For marks 88,

$$\textit{min-max} = \frac{88-88}{94-88} \times (1-0) + 0 = 0$$

Similarly, other marks can be computed as follows:

For marks 90,

$$\textit{min-max} = \frac{90-88}{94-88} \times (1-0) + 0 = 0.33$$

For marks 92,

$$\textit{min-max} = \frac{92-88}{94-88} \times (1-0) + 0 = \frac{4}{6} = 0.66$$

For marks 94,

$$\textit{min-max} = \frac{94-88}{94-88} \times (1-0) + 0 = \frac{6}{6} = 1$$

So, it can be observed that the marks {88, 90, 92, 94} are mapped to the new range {0, 0.33, 0.66, 1}. Thus, the Min-Max normalization range is between 0 and 1.

z-Score Normalization This procedure works by taking the difference between the field value and mean value, and by scaling this difference by standard deviation of the attribute.

$$V* = V - \mu/\sigma \qquad (2.2)$$

Here, s is the standard deviation of the list V and m is the mean of the list V.
Example 2.3: Consider the mark list V = {10, 20, 30}, convert the marks to z-score.
Solution: The mean and Sample Standard deviation (s) values of the list V are 20 and 10, respectively. So the z-scores of these marks are calculated using Eq. (2.2) as:

$$z\text{-score of } 10 = \frac{10-20}{10} = -\frac{10}{10} = -1$$

$$z\text{-score of } 20 = \frac{20-20}{10} = \frac{0}{10} = 0$$

$$z\text{-score of } 30 = \frac{30-20}{10} = \frac{10}{10} = 1$$

Hence, the z-score of the marks 10, 20, 30 are -1, 0 and 1, respectively.

Data Reduction

Data reduction reduces data size but produces the same results. There are different ways in which data reduction can be carried out such as data aggregation, feature selection, and dimensionality reduction.

2.4 DESCRIPTIVE STATISTICS

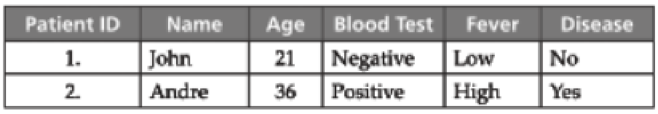
Descriptive statistics is a branch of statistics that does dataset summarization. It is used to summarize and describe data. Descriptive statistics are just descriptive and do not go beyond that. In other words, descriptive statistics do not bother too much about machine learning algorithms and its functioning.
Let us discuss descriptive statistics with the fundamental concepts of datatypes.

Dataset and Data Types

A dataset can be assumed to be a collection of data objects. The data objects may be records, points, vectors, patterns, events, cases, samples or observations. These records contain many attributes. An attribute can be defined as the property or characteristics of an object.

For example, consider the following database shown in sample Table 2.2.

Table 2.2: Sample Patient Table



Patient ID	Name	Age	Blood Test	Fever	Disease
1.	John	21	Negative	Low	No
2.	Andre	36	Positive	High	Yes

Every attribute should be associated with a value. This process is called measurement.
The type of attribute determines the data types, often referred to as measurement scale types.
The data types are shown in Figure 2.1.

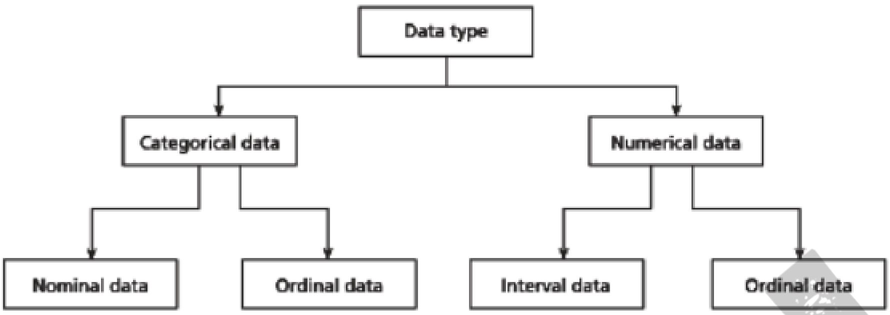


Figure 2.1: Types of Data

Broadly, data can be classified into two types:
1. Categorical or qualitative data
2. Numerical or quantitative data

Categorical or Qualitative Data The categorical data can be divided into two types. They are nominal type and ordinal type.

- •Nominal Data – In Table 2.2, patient ID is nominal data. Nominal data are symbols and cannot be processed like a number. For example, the average of a patient ID does not make any statistical sense. Nominal data type provides only information but has no ordering among data. Only operations like (=, ≠) are meaningful for these data. For example, the patient ID can be checked for equality and nothing else.
- •Ordinal Data – It provides enough information and has natural order. For example, Fever = {Low, Medium, High} is an ordinal data. Certainly, low is less than medium and medium is less than high, irrespective of the value. Any transformation can be applied to these data to get a new value.

Numeric or Qualitative Data It can be divided into two categories. They are interval type and ratio type.

- •Interval Data – Interval data is a numeric data for which the differences between values are meaningful. For example, there is a difference between 30 degree and 40 degree. Only the permissible operations are + and -.
- •Ratio Data – For ratio data, both differences and ratio are meaningful. The difference between the ratio and interval data is the position of zero in the scale. For example, take the Centigrade-Fahrenheit conversion. The zeroes of both scales do not match. Hence, these are interval data.

Another way of classifying the data is to classify it as:
1.Discrete value data
2.Continuous data

Discrete Data This kind of data is recorded as integers. For example, the responses of the survey can be discrete data. Employee identification number such as 10001 is discrete data.

Continuous Data It can be fitted into a range and includes decimal point. For example, age is a continuous data. Though age appears to be discrete data, one may be 12.5 years old and it makes sense. Patient height and weight are all continuous data.

Third way of classifying the data is based on the number of variables used in the dataset. Based on that, the data can be classified as univariate data, bivariate data, and multivariate data. This is shown in Figure 2.2.

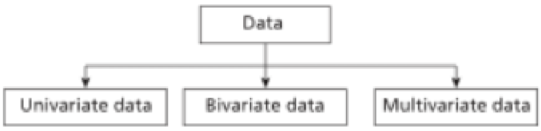


Figure 2.2: Types of Data Based on Variables

2.5 UNIVARIATE DATA ANALYSIS AND VISUALIZATION

Univariate analysis is the simplest form of statistical analysis. As the name indicates, the dataset has only one variable. A variable can be called as a category. Univariate does not deal with cause or relationships. The aim of univariate analysis is to describe data and find patterns.

Univariate data description involves finding the frequency distributions, central tendency measures, dispersion or variation, and shape of the data.

2.5.1 Data Visualization

Let us consider some forms of graphs

Bar Chart A Bar chart (or Bar graph) is used to display the frequency distribution for variables. Bar charts are used to illustrate discrete data. The charts can also help to explain the counts of nominal data. It also helps in comparing the frequency of different groups.

The bar chart for students' marks {45, 60, 60, 80, 85} with Student ID = {1, 2, 3, 4, 5} is shown below in Figure 2.3.

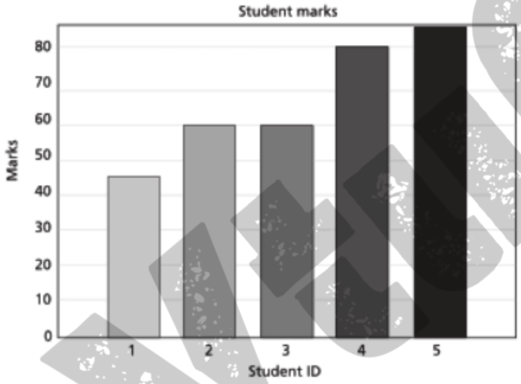


Figure 2.3: Bar Chart

Pie Chart These are equally helpful in illustrating the univariate data. The percentage frequency distribution of students' marks {22, 22, 40, 40, 70, 70, 70, 85, 90, 90} is below in Figure 2.4.

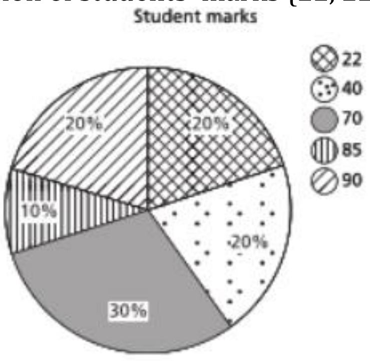


Figure 2.4: Pie Chart

It can be observed that the number of students with 22 marks are 2. The total number of students are 10. So, 2/10 × 100 = 20% space in a pie of 100% is allotted for marks 22 in Figure 2.4.

Histogram It plays an important role in data mining for showing frequency distributions. The histogram for students' marks {45, 60, 60, 80, 85} in the group range of 0-25, 26-50, 51-75, 76-100 is given below in Figure 2.5. One can visually inspect from Figure 2.5 that the number of students in the range 76-100 is 2.

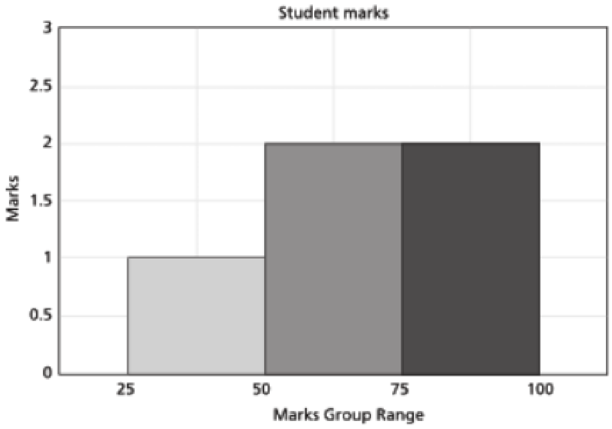


Figure 2.5: Sample Histogram of English Marks

Histogram conveys useful information like nature of data and its mode. Mode indicates the peak of dataset. In other words, histograms can be used as charts to show frequency, skewness present in the data, and shape.

Dot Plots These are similar to bar charts. They are less clustered as compared to bar charts, as they illustrate the bars only with single points. The dot plot of English marks for five students with ID as {1, 2, 3, 4, 5} and marks {45, 60, 60, 80, 85} is given in Figure 2.6. The advantage is that by visual inspection one can find out who got more marks.

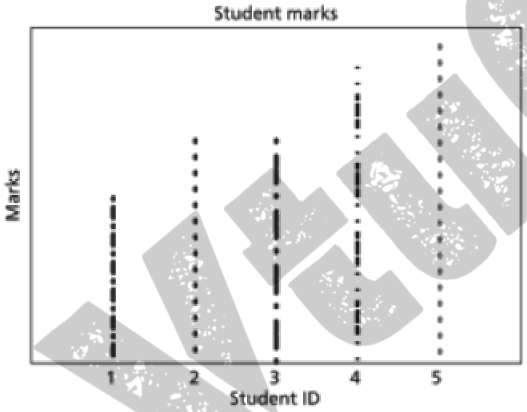


Figure 2.6: Dot Plots

2.5.2 Central Tendency

Therefore, a condensation or summary of the data is necessary. This makes the data analysis easy and simple. One such summary is called central tendency. Thus, central tendency can explain the characteristics of data and that further helps in comparison. Mass data have tendency to concentrate at certain values, normally in the central location. It is called measure of central tendency (or averages). Popular measures are mean, median and mode.

1. Mean – Arithmetic average (or mean) is a measure of central tendency that represents the 'center' of the dataset. Mathematically, the average of all the values in the sample (population) is denoted as x. Let x1, x2, … , xN be a set of 'N' values or observations, then the arithmetic mean is given as:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_N}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (2.3)$$

For example, the mean of the three numbers 10, 20, and 30 is 20

•Weighted mean – Unlike arithmetic mean that gives the weightage of all items equally, weighted mean gives different importance to all items as the item importance varies. Hence, different weightage can be given to items. In case of frequency distribution, mid values of the range are taken for computation. This is illustrated in the following computation.

In weighted mean, the mean is computed by adding the product of proportion and group mean. It is mostly used when the sample sizes are unequal.

•Geometric mean – Let x1, x2, … , xN be a set of ‘N’ values or observations. Geometric mean is the Nth root of the product of N items. The formula for computing geometric mean is given as follows:

$$\text{Geometric mean} = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{N}} = \sqrt[N]{x_1 \times x_2 \times \cdots \times x_N} \qquad (2.4)$$

Here, n is the number of items and xi are values. For example, if the values are 6 and 8, the geometric mean is given as In larger cases, computing geometric mean is difficult. Hence, it is usually calculated as:

$$\text{Anti-log of } \frac{\log(x_1) + \log(x_2) + \cdots + \log(x_N)}{N} \qquad (2.5)$$

$$= \text{anti-log } \frac{\sum_{i=1}^{n} \log(x_i)}{N} \qquad (2.6)$$

The problem of mean is its extreme sensitiveness to noise. Even small changes in the input affect the mean drastically. Hence, often the top 2% is chopped off and then the mean is calculated for a larger dataset.

2. Median – The middle value in the distribution is called median. If the total number of items in the distribution is odd, then the middle value is called median. A median class is that class where (N/2)th item is present.
In the continuous case, the median is given by the formula:

$$\text{Median} = L_1 + \frac{\frac{N}{2} - cf}{f} \times i \qquad (2.7)$$

Median class is that class where N/2th item is present. Here, i is the class interval of the median class and L1 is the lower limit of median class, f is the frequency of the median class, and cf is the cumulative frequency of all classes preceding median.
3. Mode – Mode is the value that occurs more frequently in the dataset. In other words, the value that has the highest frequency is called mode.

2.5.3 Dispersion

The spreadout of a set of data around the central tendency (mean, median or mode) is called dispersion. Dispersion is represented by various ways such as range, variance, standard deviation, and standard error. These are second order measures. The most common measures of the dispersion data are listed below:

Range Range is the difference between the maximum and minimum of values of the given list of data.

Standard Deviation The mean does not convey much more than a middle point. For example, the following datasets {10, 20, 30} and {10, 50, 0} both have a mean of 20. The difference between these two sets is the spread of data. Standard deviation is the average distance from the mean of the dataset to each point.

The formula for sample standard deviation is given by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \qquad (2.8)$$

Here, N is the size of the population, xi is observation or value from the population and m is the population mean. Often, N – 1 is used instead of N in the denominator of Eq. (2.8).

Quartiles and Inter Quartile Range It is sometimes convenient to subdivide the dataset using coordinates. Percentiles are about data that are less than the coordinates by some percentage of the total value. kth percentile is the property that the k% of the data lies at or below Xi. For example, median is 50th percentile and can be denoted as Q0.50. The 25th percentile is called first quartile (Q1) and the 75th percentile is called third quartile (Q3). Another measure that is useful to measure dispersion is Inter Quartile Range (IQR). The IQR is the difference between Q3 and Q1.
Interquartile percentile = Q3 – Q1 (2.9)
Outliers are normally the values falling apart at least by the amount 1.5 × IQR above the third quartile or below the first quartile.
Interquartile is defined by Q0.75 – Q0.25. (2.10)

Example 2.4: For patients' age list {12, 14, 19, 22, 24, 26, 28, 31, 34}, find the IQR.
Solution: The median is in the fifth position. In this case, 24 is the median. The first quartile is median of the scores below the mean i.e., {12, 14, 19, 22}. Hence, it's the median of the list below 24. In this case, the median is the average of the second and third values, that is, Q0.25 = 16.5. Similarly, the third quartile is the median of the values above the median, that is {26, 28, 31, 34}. So, Q0.75 is the average of the seventh and eighth score. In this case, it is 28 + 31/2 = 59/2 = 29.5.
Hence, the IQR using Eq. (2.10) is:
= Q0.75 – Q0.25
= 29.5-16.5 = 13

Five-point Summary and Box Plots The median, quartiles Q1 and Q3, and minimum and maximum written in the order < Minimum, Q1, Median, Q3, Maximum > is known as five-point summary.

Example 2.5: Find the 5-point summary of the list {13, 11, 2, 3, 4, 8, 9}.
Solution: The minimum is 2 and the maximum is 13. The Q1, Q2 and Q3 are 3, 8 and 11, respectively. Hence, 5-point summary is {2, 3, 8, 11, 13}, that is, {minimum, Q1, median, Q3, maximum}. Box plots are useful for describing 5-point summary. The Box plot for the set is given in Figure 2.7.

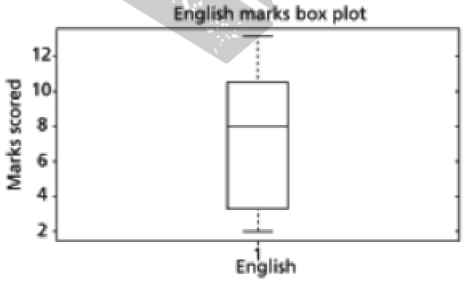


Figure 2.7: Box Plot for English Marks

2.5.4 Shape

Skewness and Kurtosis (called moments) indicate the symmetry/asymmetry and peak location of the dataset.

Skewness

The measures of direction and degree of symmetry are called measures of third order. Ideally, skewness should be zero as in ideal normal distribution. More often, the given dataset may not have perfect symmetry (consider the following Figure 2.8).

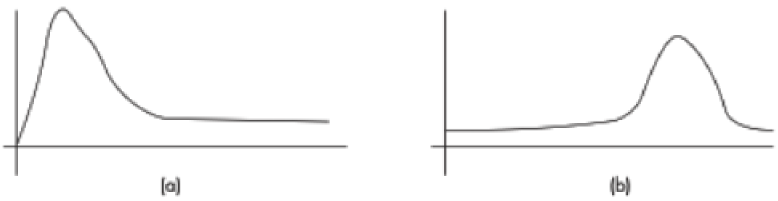


Figure 2.8: (a) Positive Skewed and (b) Negative Skewed Data

Generally, for negatively skewed distribution, the median is more than the mean. The relationship between skew and the relative size of the mean and median can be summarized by a convenient numerical skew index known as Pearson 2 skewness coefficient.

$$\frac{3 \times (\mu - median)}{\sigma} \qquad (2.12)$$

Also, the following measure is more commonly used to measure skewness. Let X1, X2, ..., XN be a set of 'N' values or observations then the skewness can be given as:

$$\frac{1}{N} \times \sum_{i=1}^{N} \frac{(x_i - \mu)^3}{\sigma^3} \qquad (2.13)$$

Here, m is the population mean and s is the population standard deviation of the univariate data. Sometimes, for bias correction instead of N, N - 1 is used.

Kurtosis

Kurtosis also indicates the peaks of data. If the data is high peak, then it indicates higher kurtosis and vice versa. Kurtosis is measured using the formula given below:

$$\frac{\sum_{i=1}^{N}(x_i - \bar{x})^4 / N}{\sigma^4} \qquad (2.14)$$

It can be observed that N - 1 is used instead of N in the numerator of Eq. (2.14) for bias correction. Here, x and s are the mean and standard deviation of the univariate data, respectively.
Some of the other useful measures for finding the shape of the univariate dataset are mean absolute deviation (MAD) and coefficient of variation (CV).

Mean Absolute Deviation (MAD)

MAD is another dispersion measure and is robust to outliers. Normally, the outlier point is detected by computing the deviation from median and by dividing it by MAD. Here, the absolute deviation between the data and mean is taken. Thus, the absolute deviation is given as:

$$|x - \mu| \qquad (2.15)$$

The sum of the absolute deviations is given as $\Sigma |x - \mu|$

Therefore, the mean absolute deviation is given as: $\frac{\Sigma|x - \mu|}{N}$ (2.16)

Coefficient of Variation (CV)

Coefficient of variation is used to compare datasets with different units. CV is the ratio of standard deviation and mean, and %CV is the percentage of coefficient of variations.

2.5.5 Special Univariate Plots

The ideal way to check the shape of the dataset is a stem and leaf plot. A stem and leaf plot are a display that help us to know the shape and distribution of the data. In this method, each value is split into a 'stem' and a 'leaf'. The last digit is usually the leaf and digits to the left of the leaf mostly form the stem. For example, marks 45 are divided into stem 4 and leaf 5 in Figure 2.9.
The stem and leaf plot for the English subject marks, say, {45, 60, 60, 80, 85} is given in

Figure 2.9.

Stem	Leaf
4	5
5	
6	0 0
7	
8	0 5

Figure 2.9: Stem and Leaf Plot for English Marks

It can be seen from Figure 2.9 that the first column is stem and the second column is leaf.
For the given English marks, two students with 60 marks are shown in stem and leaf plot as stem-6 with 2 leaves with 0. The normal Q-Q plot for marks x = [13 11 2 3 4 8 9] is given below in Figure 2.10.

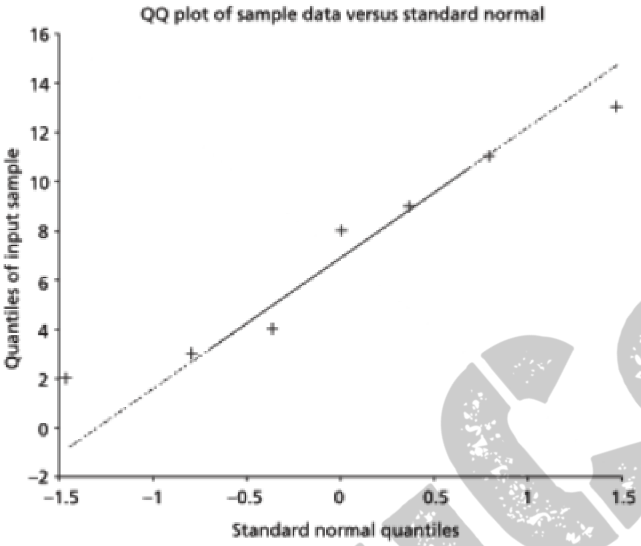


Figure 2.10: Normal *Q-Q* Plot

2.6 BIVARIATE DATA AND MULTIVARIATE DATA

Bivariate Data involves two variables. Bivariate data deals with causes of relationships. The aim is to find relationships among data. Consider the following Table 2.3, with data of the temperature in a shop and sales of sweaters.

Table 2.3: Temperature in a Shop and Sales Data

Temperature (in centigrade)	Sales of Sweaters (in thousands)
5	200
10	150
15	140
20	75
22	60
23	55
25	20

Here, the aim of bivariate analysis is to find relationships among variables. The relationships can then be used in comparisons, finding causes, and in further explorations. To do that, graphical display of the data is necessary. One such graph method is called scatter plot.

Scatter plot is used to visualize bivariate data. It is useful to plot two variables with or without nominal variables, to illustrate the trends, and also to show differences. It is a plot between explanatory and response variables. It is a 2D graph showing the relationship between two variables.

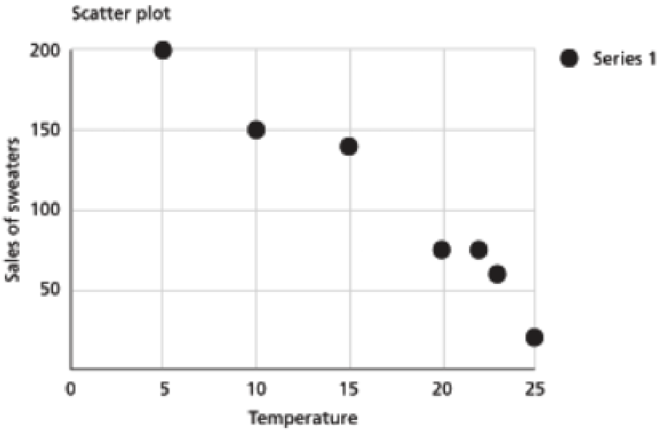


Figure 2.11: Scatter Plot

Line graphs are similar to scatter plots. The Line Chart for sales data is shown in Figure 2.12.

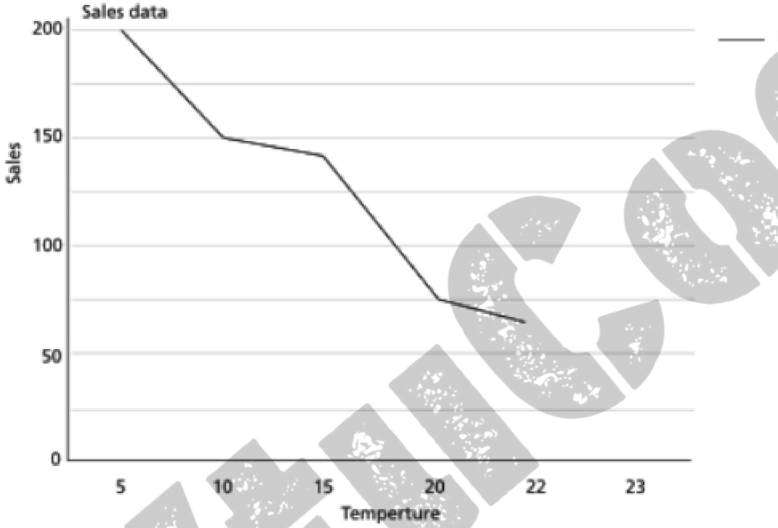


Figure 2.12: Line Chart

2.6.1 Bivariate Statistics

Covariance and Correlation are examples of bivariate statistics. Covariance is a measure of joint probability of random variables, say X and Y. Generally, random variables are represented in capital letters. It is defined as covariance(X, Y) or COV(X, Y) and is used to measure the variance between two dimensions. The formula for finding co-variance for specific x, and y are:

$$\text{cov}(X, Y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(X))(y_i - E(Y)) \qquad (2.17)$$

Here, xi and yi are data values from X and Y. E(X) and E(Y) are the mean values of xi and yi. N is the number of given data. Also, the COV(X, Y) is same as COV(Y, X).

Example 2.6: Find the covariance of data X = {1, 2, 3, 4, 5} and Y = {1, 4, 9, 16, 25}.

Solution: Mean$(X) = E(X) = \frac{15}{5} = 3$, Mean$(Y) = E(Y) = \frac{55}{5} = 11$. The covariance is computed using Eq. (2.17) as:

$$\frac{(1-3)(1-11) + (2-3)(4-11) + (3-30)(9-11) + (4-3)(16-11) + (5-3)(25-11)}{5} = 12$$

The covariance between X and Y is 12. It can be normalized to a value between -1 and +1. This is done by dividing it by the correlation of variables. This is called Pearson correlation coefficient.

Sometimes, N - 1 is also can be used instead of N. In that case, the covariance is 60/4 = 15.

Correlation

The Pearson correlation coefficient is the most common test for determining any association between two phenomena. It measures the strength and direction of a linear relationship between the x and y variables.

1.If the value is positive, it indicates that the dimensions increase together.

2.If the value is negative, it indicates that while one-dimension increases, the other dimension decreases.

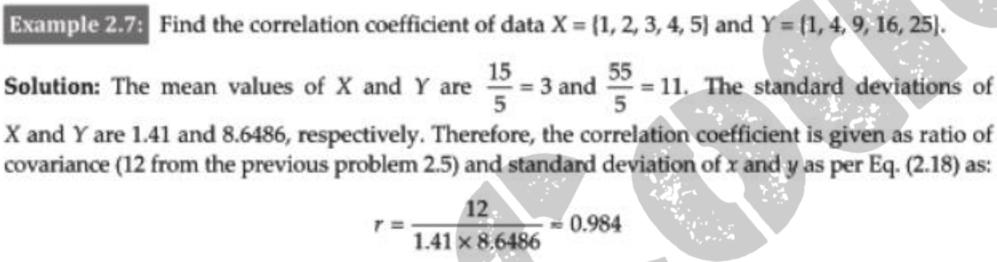
3.If the value is zero, then it indicates that both the dimensions are independent of each other.

If the dimensions are correlated, then it is better to remove one dimension as it is a redundant dimension.

If the given attributes are X = (x1, x2, … , xN) and Y = (y1, y2, … , yN), then the Pearson correlation coefficient, that is denoted as r, is given as:

$$r = \frac{COV(X,Y)}{\sigma_X \sigma_Y} \qquad (2.18)$$

where, sX, sY are the standard deviations of X and Y.



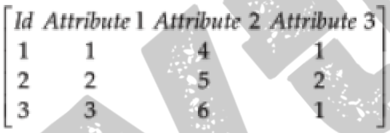
Example 2.7: Find the correlation coefficient of data X = {1, 2, 3, 4, 5} and Y = {1, 4, 9, 16, 25}.

Solution: The mean values of X and Y are $\frac{15}{5} = 3$ and $\frac{55}{5} = 11$. The standard deviations of X and Y are 1.41 and 8.6486, respectively. Therefore, the correlation coefficient is given as ratio of covariance (12 from the previous problem 2.5) and standard deviation of x and y as per Eq. (2.18) as:

$$r = \frac{12}{1.41 \times 8.6486} = 0.984$$

2.7 MULTIVARIATE STATISTICS

In machine learning, almost all datasets are multivariable. Multivariate data is the analysis of more than two observable variables, and often, thousands of multiple measurements need to be conducted for one or more subjects.



$$\begin{bmatrix} Id & Attribute\ 1 & Attribute\ 2 & Attribute\ 3 \\ 1 & 1 & 4 & 1 \\ 2 & 2 & 5 & 2 \\ 3 & 3 & 6 & 1 \end{bmatrix}$$

Multivariate data has three or more variables. The aim of the multivariate analysis is much more. They are regression analysis, factor analysis and multivariate analysis of variance that are explained in the subsequent chapters of this book.

Heatmap

Heatmap is a graphical representation of 2D matrix. It takes a matrix as input and colours it. The darker colours indicate very large values and lighter colours indicate smaller values. The advantage of this method is that humans perceive colours well. So, by colour shaping, larger values can be perceived well. For example, in vehicle traffic data, heavy traffic regions can be differentiated from low traffic regions through heatmap.

In Figure 2.13, patient data highlighting weight and health status is plotted. Here, X-axis is weights and Y-axis is patient counts. The dark colour regions highlight patients' weights vs patient counts in health status.

$$= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}.$$

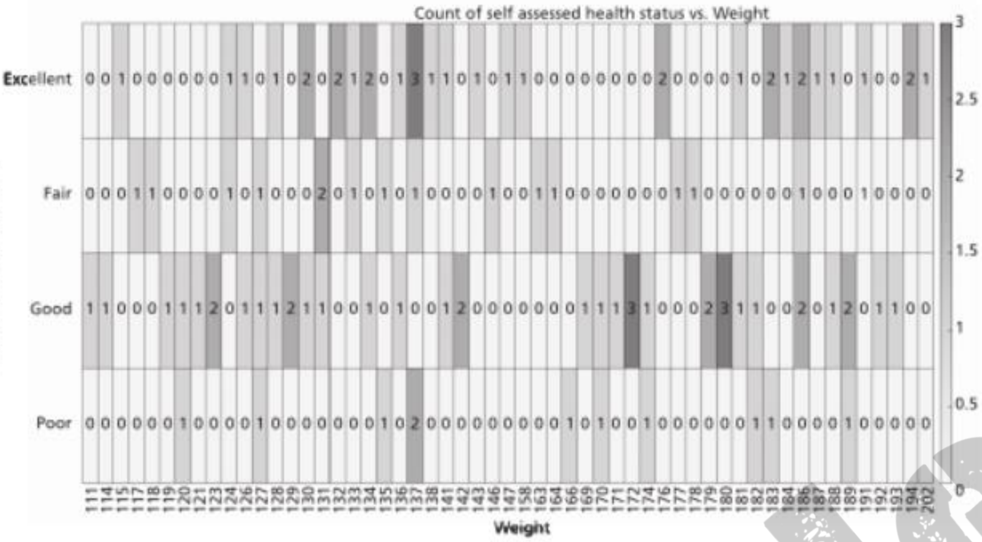


Figure 2.13: Heatmap for Patient Data

Pairplot

Pairplot or scatter matrix is a data visual technique for multivariate data. A scatter matrix consists of several pair-wise scatter plots of variables of the multivariate data.
A random matrix of three columns is chosen and the relationships of the columns is plotted as a pairplot (or scatter matrix) as shown below in Figure 2.14.

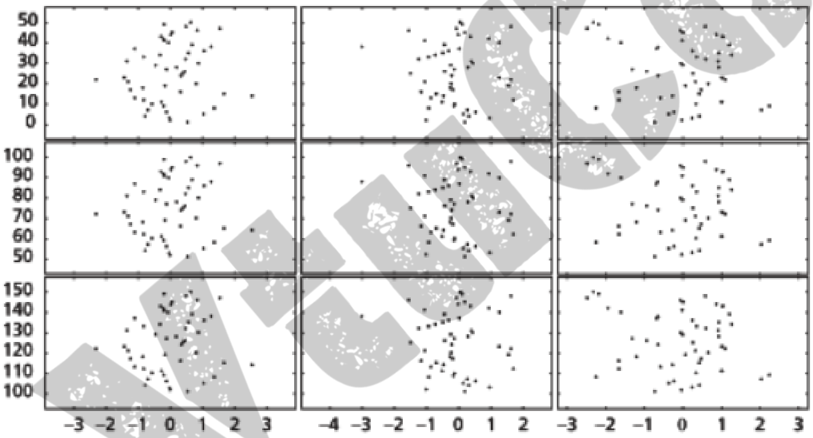


Figure 2.14: Pairplot for Random Data

2.8 ESSENTIAL MATHEMATICS FOR MULTIVARIATE DATA

Machine learning involves many mathematical concepts from the domain of Linear algebra, Statistics, Probability and Information theory. The subsequent sections discuss important aspects of linear algebra and probability.

2.8.1 Linear Systems and Gaussian Elimination for Multivariate Data

A linear system of equations is a group of equations with unknown variables.
Let Ax = y, then the solution x is given as:

$$x = y/A = A^{-1} y \qquad (2.19)$$

This is true if y is not zero and A is not zero. The logic can be extended for N-set of equations with 'n' unknown variables.
It means if A= and y=(y1 y2…yn), then the unknown variable x can be

computed as:

$$x = y/A = A^{-1}\,y \qquad (2.20)$$

If there is a unique solution, then the system is called consistent independent. If there are various solutions, then the system is called consistent dependant. If there are no solutions and if the equations are contradictory, then the system is called inconsistent.
For solving large number of system of equations, Gaussian elimination can be used. The procedure for applying Gaussian elimination is given as follows:
1.Write the given matrix.
2.Append vector y to the matrix A. This matrix is called augmentation matrix.
3.Keep the element a11 as pivot and eliminate all a11 in second row using the matrix operation,

$R_2 - \left(\frac{a_{21}}{a_{11}}\right)$, here R_2 is the second row and $\left(\frac{a_{21}}{a_{11}}\right)$ is called the multiplier.

The same logic
can be used to remove a11 in all other equations.
4.Repeat the same logic and reduce it to reduced echelon form. Then, the unknown variable as:

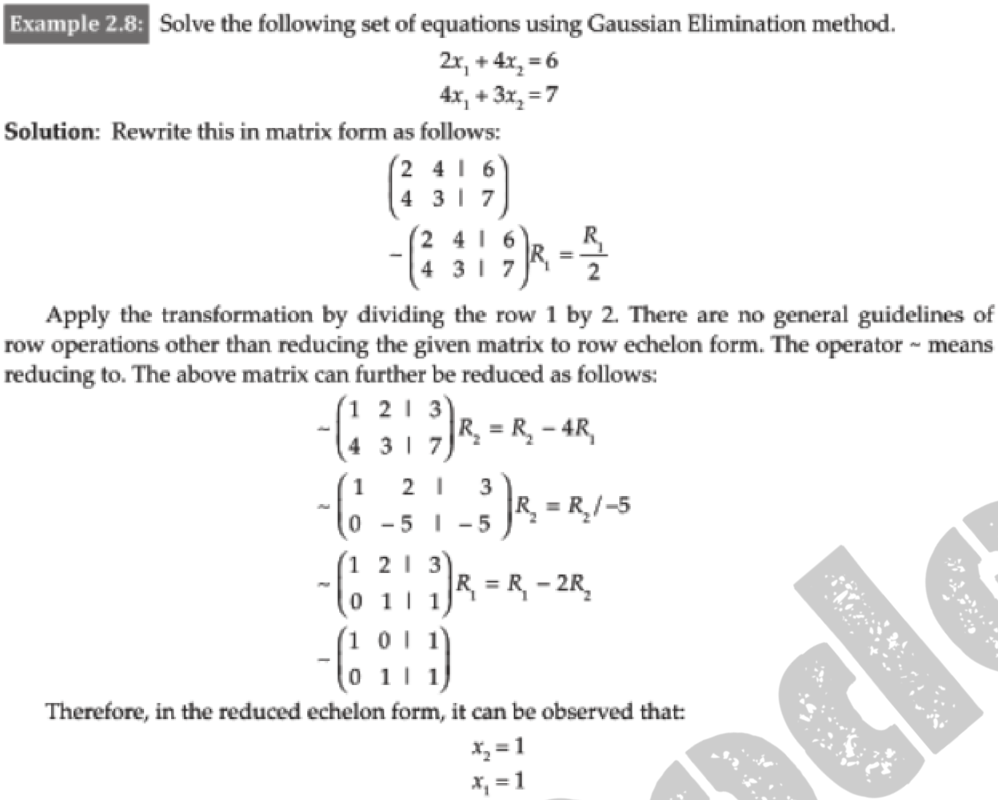
$$x_n = \frac{y_{nn}}{a_{nn}} \qquad (2.21)$$

5.Then, the remaining unknown variables can be found by back-substitution as:

$$x_{n-1} = \frac{y_{n-1} - a_{n-1} \times x_n}{a_{(n-1)(n-1)}} \qquad (2.22)$$

This part is called backward substitution.

To facilitate the application of Gaussian elimination method, the following row operations are applied:
1.Swapping the rows
2.Multiplying or dividing a row by a constant
3.Replacing a row by adding or subtracting a multiple of another row to it
These concepts are illustrated in Example 2.8.



Example 2.8: Solve the following set of equations using Gaussian Elimination method.

$$2x_1 + 4x_2 = 6$$
$$4x_1 + 3x_2 = 7$$

Solution: Rewrite this in matrix form as follows:

$$\begin{pmatrix} 2 & 4 & | & 6 \\ 4 & 3 & | & 7 \end{pmatrix}$$

$$\sim \begin{pmatrix} 2 & 4 & | & 6 \\ 4 & 3 & | & 7 \end{pmatrix} R_1 = \frac{R_1}{2}$$

Apply the transformation by dividing the row 1 by 2. There are no general guidelines of row operations other than reducing the given matrix to row echelon form. The operator ~ means reducing to. The above matrix can further be reduced as follows:

$$\sim \begin{pmatrix} 1 & 2 & | & 3 \\ 4 & 3 & | & 7 \end{pmatrix} R_2 = R_2 - 4R_1$$

$$\sim \begin{pmatrix} 1 & 2 & | & 3 \\ 0 & -5 & | & -5 \end{pmatrix} R_2 = R_2 / -5$$

$$\sim \begin{pmatrix} 1 & 2 & | & 3 \\ 0 & 1 & | & 1 \end{pmatrix} R_1 = R_1 - 2R_2$$

$$\sim \begin{pmatrix} 1 & 0 & | & 1 \\ 0 & 1 & | & 1 \end{pmatrix}$$

Therefore, in the reduced echelon form, it can be observed that:

$$x_2 = 1$$
$$x_1 = 1$$

2.8.2 Matrix Decomposition

It is often necessary to reduce a matrix to its constituent parts so that complex matrix operations can be performed.

Then, the matrix A can be decomposed as:

$$A = Q \Lambda Q^T \qquad (2.23)$$

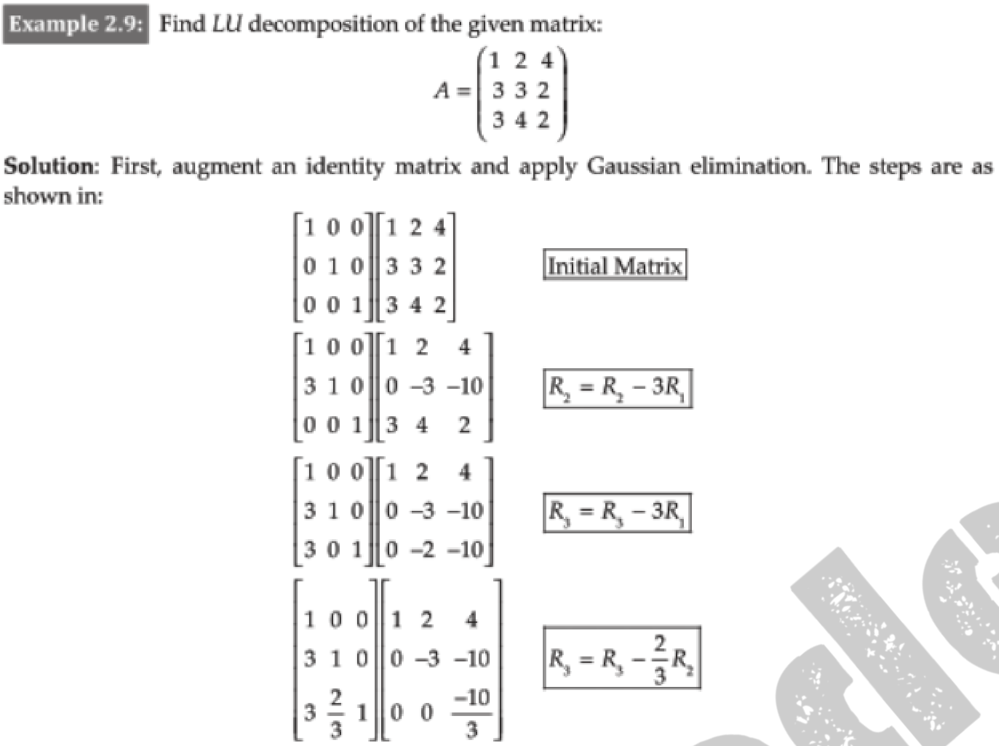
where, Q is the matrix of eigen vectors, Λ is the diagonal matrix and QT is the transpose of matrix Q.

LU Decomposition

One of the simplest matrix decomposition is LU decomposition where the matrix A can be decomposed matrices: A = LU

Here, L is the lower triangular matrix and U is the upper triangular matrix. The decomposition can be done using Gaussian elimination method as discussed in the previous section. First, an identity matrix is augmented to the given matrix. Then, row operations and Gaussian elimination is applied to reduce the given matrix to get matrices L and U.

Example 2.9 illustrates the application of Gaussian elimination to get LU.



Example 2.9: Find LU decomposition of the given matrix:

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{pmatrix}$$

Solution: First, augment an identity matrix and apply Gaussian elimination. The steps are as shown in:

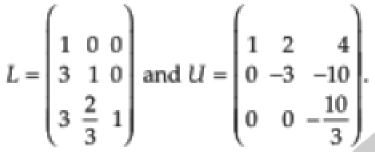
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{bmatrix} \quad \boxed{\text{Initial Matrix}}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 3 & 4 & 2 \end{bmatrix} \quad \boxed{R_2 = R_2 - 3R_1}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & -2 & -10 \end{bmatrix} \quad \boxed{R_3 = R_3 - 3R_1}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & \frac{2}{3} & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & 0 & \frac{-10}{3} \end{bmatrix} \quad \boxed{R_3 = R_3 - \frac{2}{3}R_2}$$

Now, it can be observed that the first matrix is L as it is the lower triangular matrix whose values are the determiners used in the reduction of equations above such as 3, 3 and 2/3.
The second matrix is U, the upper triangular matrix whose values are the values of the reduced matrix because of Gaussian elimination.



$$L = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & \frac{2}{3} & 1 \end{pmatrix} \text{ and } U = \begin{pmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & 0 & -\frac{10}{3} \end{pmatrix}.$$

2.8.3 Machine Learning and Importance of Probability and Statistics

Machine learning is linked with statistics and probability. Like linear algebra, statistics is the heart of machine learning. The importance of statistics needs to be stressed as without statistics;

Probability Distributions

A probability distribution of a variable, say X, summarizes the probability associated with X's events. Distribution is a parameterized mathematical function. In other words, distribution is a function that describes the relationship between the observations in a sample space.
Consider a set of data. The data is said to follow a distribution if it obeys a mathematical function that characterizes that distribution. The function can be used to calculate the probability of individual observations.
Probability distributions are of two types:
1.Discrete probability distribution
2.Continuous probability distribution
The relationships between the events for a continuous random variable and their probabilities

Continuous Probability Distributions Normal, Rectangular, and Exponential distributions fall under this category.

1. **Normal Distribution** – Normal distribution is a continuous probability distribution. This is also known as gaussian distribution or bell-shaped curve distribution. It is the most common distribution function. The shape of this distribution is a typical bell-shaped

curve. In normal distribution, data tends to be around a central value with no bias on left or right. The heights of the students, blood pressure of a population, and marks scored in a class can be approximated using normal distribution.

PDF of the normal distribution is given as:

$$f(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (2.24)$$

Here, m is mean and s is the standard deviation. Normal distribution is characterized by two parameters – mean and variance.

One important concept associated with normal distribution is z-score. It can be computed as:

$z = \frac{x - \mu}{\sigma}$. When μ is zero and σ is 1, z-score is same as x.

This is useful to normalize the data.

2. **Rectangular Distribution** – This is also known as uniform distribution. It has equal probabilities for all values in the range a, b. The uniform distribution is given as follows:

$$P(X = x) = \begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b \\ 0 & \text{Otherwise} \end{cases} \qquad (2.25)$$

3.Exponential Distribution – This is a continuous uniform distribution. This probability distribution is used to describe the time between events in a Poisson process. Exponential distribution is another special case of Gamma distribution with a fixed parameter of 1. This distribution is helpful in modelling of time until an event occurs.

The PDF is given as follows:

$$f(x, \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \ge 0 \quad (\lambda > 0) \\ 0 & \text{if } x < 0 \end{cases} \qquad (2.26)$$

Here, x is a random variable and λ is called rate parameter. The mean and standard deviation of exponential distribution is given as β, where, $\beta = \frac{1}{\lambda}$.

Discrete Distribution Binomial, Poisson, and Bernoulli distributions fall under this category.

1. Binomial Distribution – Binomial distribution is another distribution that is often encountered in machine learning. It has only two outcomes: success or failure. This is also called Bernoulli trial.

The objective of this distribution is to find probability of getting success k out of n trials.

The way to get success out of k out of n number of trials is given as:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \qquad (2.27)$$

The binomial distribution function is given as follows, where p is the probability of success and probability of failure is (1 - p). The probability of success in a certain number of trials is given as:

$$p^k(1-p)^{n-k} \text{ or } p^k q^{n-k} \qquad (2.28)$$

Combining both, one gets PDF of binomial distribution as:

$$\binom{n}{k} p^k (1-p)^{n-k} \qquad (2.29)$$

Here, p is the probability of each choice, k is the number of choices, and n is the total number of choices. The mean of binomial distribution is given below:

$$\mu = n \times p \qquad (2.30)$$

And the variance is given as:

$$\sigma^2 = np(1-p) \qquad (2.31)$$

Hence, the standard deviation is given as:

$$\sigma = \sqrt{np(1-p)} \quad (2.32)$$

2. **Poisson Distribution** – It is another important distribution that is quite useful. Given an interval of time, this distribution is used to model the probability of a given number of events k. The mean rule l is inclusive of previous events. Some of the examples of Poisson distribution are number of emails received, number of customers visiting a shop and the number of phone calls received by the office.

The PDF of Poisson distribution is given as follows:

$$f(X = x; \lambda) = Pr[X = x] = \frac{e^{-\lambda}\lambda^{x}}{x!} \quad (2.33)$$

Here, x is the number of times the event occurs and λ is the mean number of times an event occurs.

The mean is the population mean at number of emails received and the standard deviation is $\sqrt{\lambda}$.

3.Bernoulli Distribution – This distribution models an experiment whose outcome is binary. The outcome is positive with p and negative with 1 - p. The PMF of this distribution is given as:

$$f(k;p) = \begin{cases} q = 1 - p & \text{if } k = 0 \\ p & \text{if } k = 1. \end{cases} \quad (2.34)$$

The mean is p and variance is p(1 - p) = q

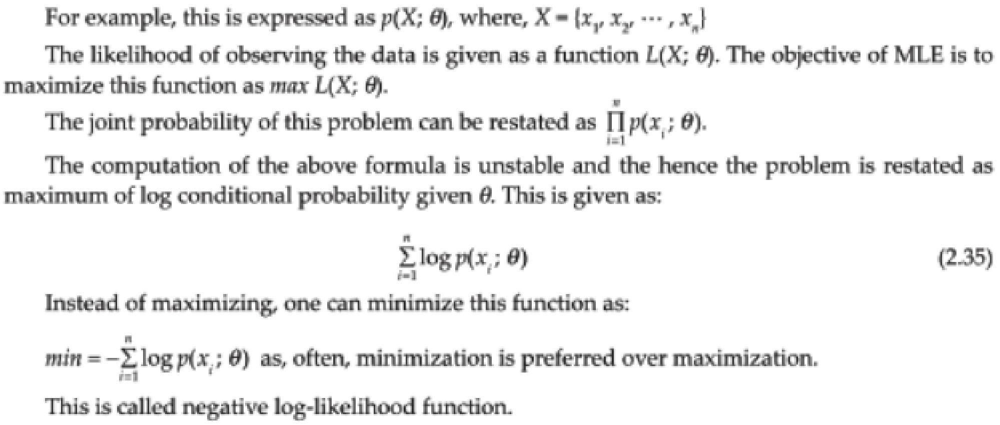
Density Estimation

Let there be a set of observed values x1, x2, … , xn from a larger set of data whose distribution is not known. Density estimation is the problem of estimating the density function from an observed data.

There are two types of density estimation methods, namely parametric density estimation and non-parametric density estimation.

Parametric Density Estimation It assumes that the data is from a known probabilistic distribution and can be estimated as $p(x \mid \Theta)$, where, Θ is the parameter. Maximum likelihood function is a parametric estimation method.

Maximum Likelihood Estimation For a sample of observations, one can estimate the probability distribution. This is called density estimation. Maximum Likelihood Estimation (MLE) is a probabilistic framework that can be used for density estimation. This involves formulating a function called likelihood function which is the conditional probability of observing the observed samples and distribution function with its parameters. For example, if the observations are X = {x1, x2, … , xn}, then density estimation is the problem of choosing a PDF with suitable parameters to describe the data. MLE treats this problem as a search or optimization problem where the probability should be maximized for the joint probabilities of X and its parameter, theta.



For example, this is expressed as $p(X; \theta)$, where, $X = \{x_1, x_2, \cdots, x_n\}$

The likelihood of observing the data is given as a function $L(X; \theta)$. The objective of MLE is to maximize this function as $max\ L(X; \theta)$.

The joint probability of this problem can be restated as $\prod_{i=1}^{n} p(x_i; \theta)$.

The computation of the above formula is unstable and the hence the problem is restated as maximum of log conditional probability given θ. This is given as:

$$\sum_{i=1}^{n} \log p(x_i; \theta) \qquad (2.35)$$

Instead of maximizing, one can minimize this function as:

$min = -\sum_{i=1}^{n} \log p(x_i; \theta)$ as, often, minimization is preferred over maximization.

This is called negative log-likelihood function.

If one assumes that the regression problem can be framed as predicting output y given input x, then for p(y/x), the MLE framework can be applied as:

$$max \sum \log(y \mid x_i, h) \qquad (2.36)$$

Here, h is the linear regression model. If Gaussian distribution is assumed as it is an obvious fact that most of the data follow Gaussian distribution, then MLE can be stated as:

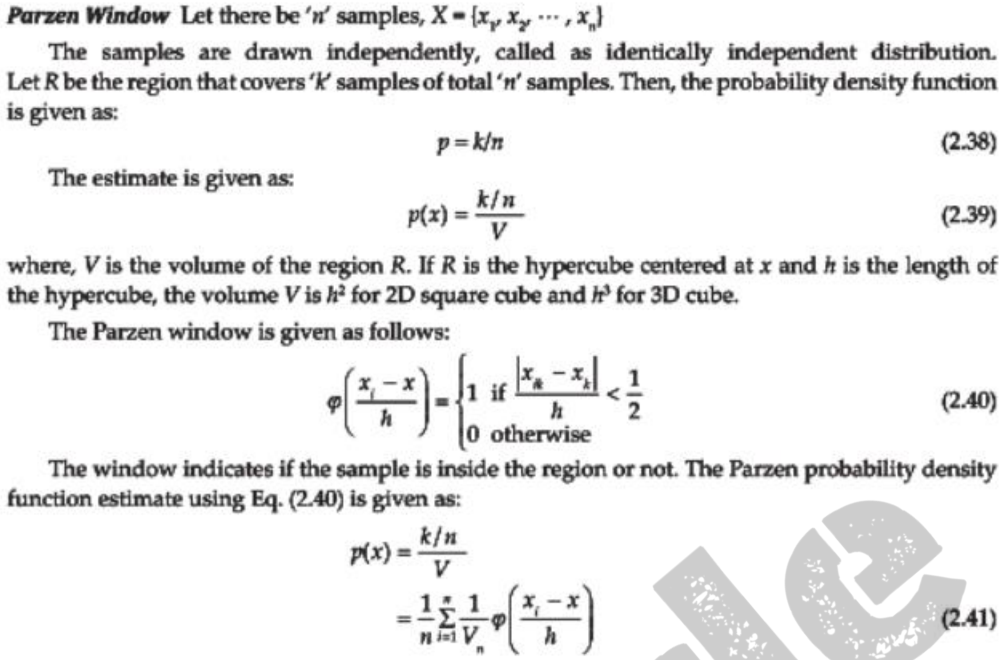
$$\max \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y_i - h(x_i;\beta)}{2\sigma^2}} \qquad (2.37)$$

Here, b is the regression coefficient and xi is the given sample. One can maximize this function or minimize the negative log likelihood function to provide a solution for linear regression problem. The Eq. (2.37) yields the same answer of the least-square approach.

Gaussian Mixture Model and Expectation-Maximization (EM) Algorithm In machine learning, clustering is one of the important tasks. It is discussed in Chapter 13. MLE framework is quite useful for designing model-based methods for clustering data. A model is a statistical method and data is assumed to be generated by a distribution model with its parameter, theta. There may be many distributions involved and that is why it is called as mixture model.

Generally, there can be many unspecified distributions with different set of parameters. The EM algorithm has two stages:
1. Expectation (E) Stage – In this stage, the expected PDF and its parameters are estimated for each latent variable.
2.Maximization (M) stage – In this, the parameters are optimized using the MLE function.
This process is iterative, and the iteration is continued till all the latent variables are fitted by probability distributions effectively along with the parameters.

Non-parametric Density Estimation A non-parametric estimation can be generative or discriminative. Parzen window is a generative estimation method that finds $p(x \mid \Theta)$ as conditional density. Discriminative methods directly compute $p(x \mid \Theta)$ as posteriori probability. Parzen window and k-Nearest Neighbour (KNN) rule are examples of non-parametric density



Parzen Window Let there be '*n*' samples, $X = \{x_1, x_2, \cdots, x_n\}$

The samples are drawn independently, called as identically independent distribution. Let R be the region that covers 'k' samples of total 'n' samples. Then, the probability density function is given as:

$$p = k/n \tag{2.38}$$

The estimate is given as:

$$p(x) = \frac{k/n}{V} \tag{2.39}$$

where, V is the volume of the region R. If R is the hypercube centered at x and h is the length of the hypercube, the volume V is h^2 for 2D square cube and h^3 for 3D cube.

The Parzen window is given as follows:

$$\varphi\left(\frac{x_i - x}{h}\right) = \begin{cases} 1 & \text{if } \dfrac{|x_k - x_k|}{h} < \dfrac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.40}$$

The window indicates if the sample is inside the region or not. The Parzen probability density function estimate using Eq. (2.40) is given as:

$$\begin{aligned} p(x) &= \frac{k/n}{V} \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{x_i - x}{h}\right) \end{aligned} \tag{2.41}$$

estimation.
This window can be replaced by any other function too. If Gaussian function is used, then it is called Gaussian density function.
KNN Estimation The KNN estimation is another non-parametric density estimation method. Here, the initial parameter k is determined and based on that k-neighbours are determined. The probability density function estimate is the average of the values that are returned by the neighbours.

2.9 OVERVIEW OF HYPOTHESIS

Data collection alone is not enough. Data must be interpreted to give a conclusion. The conclusion should be a structured outcome. This assumption of the outcome is called a hypothesis. Statistical methods are used to confirm or reject the hypothesis. The assumption of the statistical test is called null hypothesis. It is also called as hypothesis zero (H0). In other words, hypothesis is the existing belief. The violation of this hypothesis is called first hypothesis (H1) or hypothesis one. This is the hypothesis the researcher is trying to establish.
There are two types of hypothesis tests, parametric and non-parametric. Parametric tests are based on parameters such as mean and standard deviation. Non-parametric tests are dependent on characteristics such as independence of events or data following certain distribution.
Statistical tests help to:
1.Define null and alternate hypothesis
2.Describe the hypothesis using parameters
3.Identify the statistical test and statistics
4.Decide the criteria called significance value a
5.Compute p-value (probability value)
6.Take the final decision of accepting or rejecting the hypothesis based on the parameters

Hypothesis testing is particularly important as it is an integral part of the learning algorithms. Generally, the data size is small. So, one may have to know whether the hypothesis will work for additional samples and how accurate it is. No matter how effective the statistical tests are, two kinds of errors are involved, that are Type I and Type II.

Type I error is the incorrect rejection of a true null hypothesis and is called false positive.
Type II error is the incomplete failure of rejecting a false hypothesisand is called false negative.
During these calculations, one must include the size of the data sample. Degree of freedom

indicates the number of independent pieces of information used for the test. It is indicated as n. The mean or variance can be used to indicate the degree of freedom.

Hypothesis Testing

Let us define two important errors called sample error and true (or actual error). Let us assume that D is the unknown distribution, Target function is f(x): x ≥ {0, 1}, x is the instance, h(x) is the hypothesis, and sample set is S that derives the samples on instances drawn from X. Then, the actual error is denoted as:

$$error_D(h) = pr_{x\in D}\{f(x) \neq h(x)\} \qquad (2.42)$$

In other words, true error is the probability that the hypothesis will mis classify an instance that is drawn at random. The point is that population is very large and hence it is not possible to determine true error and can only be estimated. So, another error is called sample error or estimator.

Sample error is with respect to sample S. It is the probability for instances drawn from X, that is, the fractions of S that are misclassified. The sample error is given as follows:

$$error_S(h) = \frac{1}{n}\sum_{x\in S}\delta(f(x), h(x)) \qquad (2.43)$$

Here, $\delta(f(x), h(x))$ is 1 if $f(x) \neq h(x)$, otherwise is zero.

p-value

Statistical tests can be performed to either accept or reject the null hypothesis. This is done by the value called p-value or probability value. It indicates the probability of hypothesis being true. The p-value is used to interpret or quantify the test. For example, a statistical test result may give a value of 0.03. Then, one can compare it with the level 0.05. As 0.03 < 0.05, the result is assumed to be significant. This means that the variables tested are not independent. Here, 0.05 is called significant level. In general, significant level is called a and p-value is compared with a. If p-value ≤ a, then the hypothesis H1 is rejected and if p-value >a, then the hypothesis H0 is rejected.

Confidence Intervals

The acceptance or rejection of the hypothesis can also be done using confidence interval.
The confidence interval is computed as:
Confidence interval = 1 – significant level (2.44)

Confidence level is the range of values that indicates the location of true mean. Confidence intervals indicate the confidence of the result. If the confidence level is 90%, then it infers that there is 90% of chance that the true mean lies in this range and remaining 10% indicates that true mean in not present. For finding this, one requires mean and standard deviation. Then, x can be given as $mean(x) \pm z \times \frac{s}{\sqrt{N}}$. Here, s is the standard deviation, N is the number of samples, and z is the value associated with 90% and is called % of confidence. This is also called as margin of error.

Sample error is the unbiased estimate of true error. If no information is provided, then both errors are the same. It is, however, often safe to suggest a margin of confidence associated with the hypothesis. The hypothesis with 95% confidence about the sample error can be given as follows:

$$error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1 - error_S(h))}{n}} \qquad (2.45)$$

This 1.96 indicates the 95% confidence of the error. The number 1.96 can be replaced by any number that is associated with different levels of confidence.

The procedure to estimate the difference between two hypothesis, say h1 and h2, is as follows:

1.A parameter d can be chosen to estimate the error of two hypothesis:

2.d ≡ errorD(h1) - errorD(h2)(2.46)

Here, there are two hypothesis h1 and h2 tested on two sample sets s1 and s2. Similarly, n1 and n2 are randomly drawn number of samples.

3.The estimator ^d can be estimated as the difference as:

$$\hat{d} \equiv error_D(h_1) - error_D(h_2)$$

4.The confidence intervals can be used for the estimator also as follows:

$$\hat{d} \pm z_N \sqrt{\frac{error_{s_1}(h_1) - (1 - error_{s_1}(h_1))}{n_1} + \frac{error_{s_2}(h_2) - (1 - error_{s_2}(h_2))}{n_2}} \qquad (2.47)$$

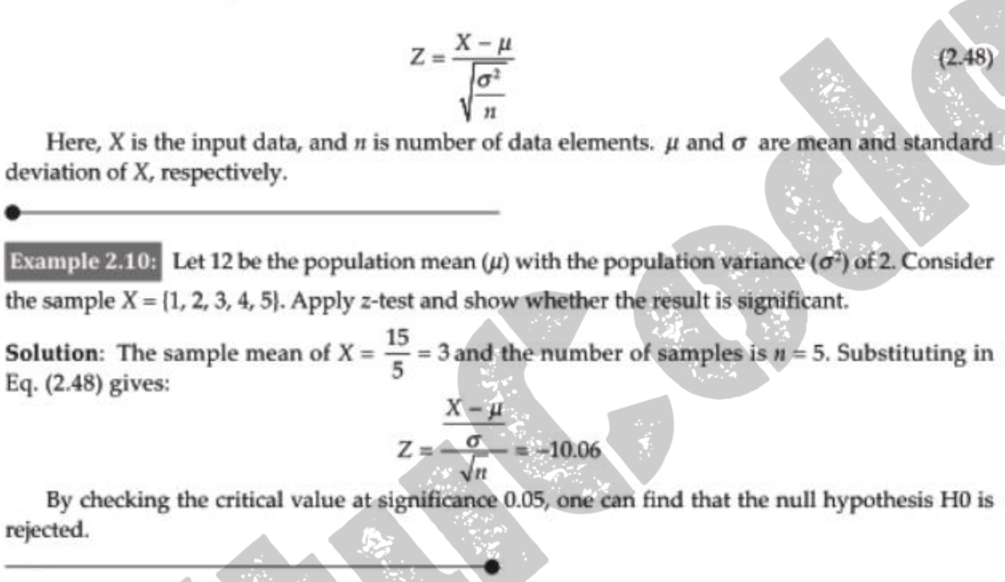
Sometimes, it is desirable to find interval L and U such that N% of the probability falls in this

2.9.1 Comparing Learning Methods

Some of the methods for comparing the learning programs are given below:

Z-test

Z-test assumes normal distribution of data whose population variation is known. The sample size is assumed to be large. The focus is to test the population mean. The z-statistic is given as:



$$Z = \frac{X - \mu}{\sqrt{\frac{\sigma^2}{n}}} \qquad (2.48)$$

Here, X is the input data, and n is number of data elements. μ and σ are mean and standard deviation of X, respectively.

Example 2.10: Let 12 be the population mean (μ) with the population variance (σ^2) of 2. Consider the sample $X = \{1, 2, 3, 4, 5\}$. Apply z-test and show whether the result is significant.

Solution: The sample mean of $X = \frac{15}{5} = 3$ and the number of samples is $n = 5$. Substituting in Eq. (2.48) gives:

$$Z = \frac{X - \mu}{\frac{\sigma}{\sqrt{n}}} = -10.06$$

By checking the critical value at significance 0.05, one can find that the null hypothesis H0 is rejected.

t-test and Paired t-test

t-test is a hypothesis test and checks if the difference between two samples' mean is real or by chance. Here, data is continuous and randomly selected. There will only be small number of samples and variance between groups is real. The t-test statistics follows t-distribution under null hypothesis and is used when the number of samples <30. So, the procedure is:

•Select a group

•Compute average

•Compare it with theoretical value and compute t-statistic:

$$t = \frac{m - \mu}{\frac{s}{\sqrt{n}}} \qquad (2.49)$$

Here, t is t-statistic, m is the mean of the group, m is the theoretical value or population mean, s is the standard deviation, and n is the group size or sample size.

Independent Two Sample t-test t-statistic for two groups A and B is computed as follows:

$$t = \frac{mean(A) - mean(B)}{\sqrt{\frac{(N_1 - 1)s_1^2 + (N_2 - 1)s_2^2}{N_1 + N_2 - 2}\left(\frac{1}{N_1} + \frac{1}{N_2}\right)}} \qquad (2.50)$$

Here, mean(A) and mean(B) are for two different samples. N1 and N2 are sample sizes of two groups A and B. s^2 is the variance of the two samples and the degree of freedom is given as

N1 + N2 - 2 . Then, t-statistic is compared with the t-critical value.

Paired t-test It is used to evaluate the hypothesis before and after intervention. The fact is that these samples are not independent. For example, consider the case of an effect of medication for a diabetic patient. The sequence is that first the sugar is tested, then the medication is done, and again the sugar test is conducted to study the effect of medication. In short, in paired t-test, the data is taken from the same subject twice. In an unpaired t-test, the samples are taken independently. In this only one group is involved. The t-statistic is computed as:

$$t = \frac{m - \mu}{\frac{s}{\sqrt{n}}}$$

Here, t is t-statistic, m is the mean of the group, m is the theoretical value or population mean, s is the standard deviation, and n is the group size or sample size.

Chi-Square Test

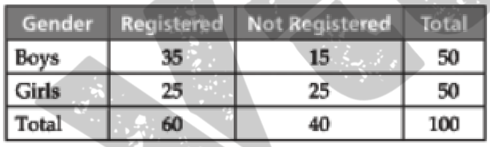
Chi-Square test is a non-parametric test. The goodness-of-fit test statistics follows a Chi-Square distribution under null hypothesis and measures the statistical significance between observed frequency and expressed frequency, and each observation is independent of each other and follows normal distribution. This comparison is used to calculate the value of the Chi-Square statistic as:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \qquad (2.51)$$

Here, E is the expected frequency, O is the observed frequency and the degree of freedom is C – 1, where, C is number of categories. The Chi-Square test allows us to detect the duplication of data and helps to remove the redundancy of values.

Example 2.11: Consider the following Table 2.4, where the machine learning course registration is done by both boys and girls. There are 50 boys and 50 girls in the class and the registration of the course is given in the table. Apply Chi-Square test and find out whether any differences exist between boys and girls for course registration.

Table 2.4: Observed Data

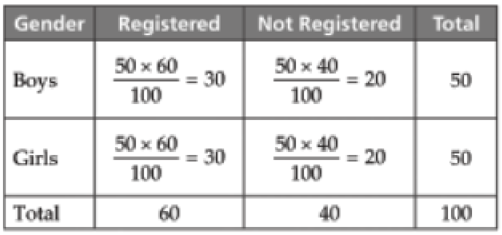


Gender	Registered	Not Registered	Total
Boys	35	15	50
Girls	25	25	50
Total	60	40	100

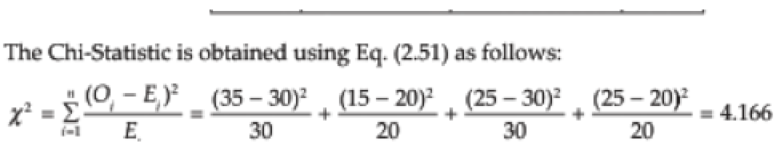
Solution: Let the null hypothesis be H0 when there is no difference between boys and girls and H1 be the alternate hypothesis when there is a significant difference between boys and girls.

For applying the Chi-Square test based on the observations, the expectation should be obtained by multiplying the total boys X registered/Total and Total girls X not registered/Total as shown in Table 2.5.

Table 2.5: Expected Data



Gender	Registered	Not Registered	Total
Boys	$\frac{50 \times 60}{100} = 30$	$\frac{50 \times 40}{100} = 20$	50
Girls	$\frac{50 \times 60}{100} = 30$	$\frac{50 \times 40}{100} = 20$	50
Total	60	40	100



The Chi-Statistic is obtained using Eq. (2.51) as follows:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} = \frac{(35-30)^2}{30} + \frac{(15-20)^2}{20} + \frac{(25-30)^2}{30} + \frac{(25-20)^2}{20} = 4.166$$

for degree of freedom = number of categories -1 = 2 - 1 = 1. The p value for this statistic is 0.0412. This is less than 0.05. Therefore, the result is significant.

2.10 FEATURE ENGINEERING AND DIMENSIONALITY REDUCTION TECHNIQUES

Features are attributes. Feature engineering is about determining the subset of features that form an important part of the input that improves the performance of the model, be it classification or any other model in machine learning.

Feature engineering deals with two problems – Feature Transformation and Feature Selection. Feature transformation is extraction of features and creating new features that may be helpful in increasing performance. For example, the height and weight may give a new attribute called Body Mass Index (BMI).

Feature subset selection is another important aspect of feature engineering that focuses on selection of features to reduce the time but not at the cost of reliability.
The features can be removed based on two aspects:
1.Feature relevancy – Some features contribute more for classification than other features. For example, a mole on the face can help in face detection than common features like nose. In simple words, the features should be relevant.
2. Feature redundancy – Some features are redundant. For example, when a database table has a field called Date of birth, then age field is not relevant as age can be computed easily from date of birth.
So, the procedure is:
1.Generate all possible subsets
2.Evaluate the subsets and model performance
3.Evaluate the results for optimal feature selection

Filter-based selection uses statistical measures for assessing features. In this approach, no learning algorithm is used. Correlation and information gain measures like mutual information and entropy are all examples of this approach.

Wrapper-based methods use classifiers to identify the best features. These are selected and evaluated by the learning algorithms. This procedure is computationally intensive but has superior performance.

2.10.1 Stepwise Forward Selection

This procedure starts with an empty set of attributes. Every time, an attribute is tested for statistical significance for best quality and is added to the reduced set. This process is continued till a good reduced set of attributes is obtained.

2.10.2 Stepwise Backward Elimination

This procedure starts with a complete set of attributes. At every stage, the procedure removes the worst attribute from the set, leading to the reduced set.

2.10.3 Principal Component Analysis

The idea of the principal component analysis (PCA) or KL transform is to transform a given set of measurements to a new set of features so that the features exhibit high information packing properties. This leads to a reduced and compact set of features.
Consider a group of random vectors of the form:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

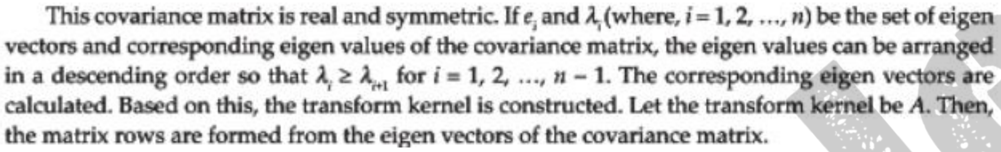
The mean vector of the set of random vectors is defined as:

$$m_x = E\{x\}$$

The operator E refers to the expected value of the population. This is calculated theoretically using the probability density functions (PDF) of the elements xi and the joint probability density functions between the elements xi and xj. From this, the covariance matrix can be calculated as:

$$m_x = \frac{1}{M}\sum_{k=1}^{M} x_k \qquad (2.53)$$

$$A = \frac{1}{M}\sum_{k=1}^{M} x_k x_k^T - m_x m_x^T \qquad (2.54)$$



This covariance matrix is real and symmetric. If e_i and λ_i (where, $i = 1, 2, \ldots, n$) be the set of eigen vectors and corresponding eigen values of the covariance matrix, the eigen values can be arranged in a descending order so that $\lambda_i \geq \lambda_{i+1}$ for $i = 1, 2, \ldots, n - 1$. The corresponding eigen vectors are calculated. Based on this, the transform kernel is constructed. Let the transform kernel be A. Then, the matrix rows are formed from the eigen vectors of the covariance matrix.

The mapping of the vectors x to y using the transformation can now be described as:

$$y = A(x - m_x) \qquad (2.55)$$

This transform is also called as Karhunen-Loeve or Hoteling transform. The original vector x can now be reconstructed as follows:

$$x = A^T y + m_x \qquad (2.56)$$

If K largest eigen values are used, the recovered information would be:

$$x = A_K^T y + m_x \qquad (2.57)$$

The PCA algorithm is as follows:
1.The target dataset x is obtained
2.The mean is subtracted from the dataset. Let the mean be m. Thus, the adjusted dataset is X – m. The objective of this process is to transform the dataset with zero mean.
3.The covariance of dataset x is obtained. Let it be C.
4.Eigen values and eigen vectors of the covariance matrix are calculated.
5.The eigen vector of the highest eigen value is the principal component of the dataset. The eigen values are arranged in a descending order. The feature vector is formed with these eigen vectors in its columns.
Feature vector = {eigen vector1, eigen vector2, … , eigen vectorn}
6.Obtain the transpose of feature vector. Let it be A.
7.PCA transform is y = A × (x – m), where x is the input dataset, m is the mean, and A is the transpose of the feature vector.
The original data can be retrieved using the formula given below:

$$\text{Original data } (f) = \{(A)^{-1} \times y\} + m \qquad (2.58)$$

$$= \{(A)^T \times y\} + m \qquad (2.59)$$

The new data is a dimensionaly reduced matrix that represents the original data.

Figure 2.15. The scree plot indicates that only 6 out of 246 attributes are important.

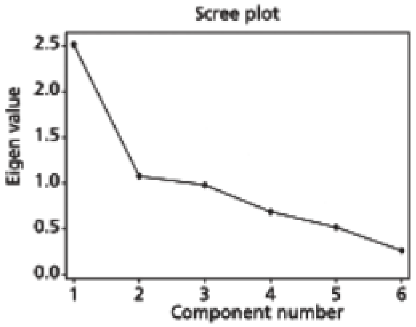


Figure 2.15: Scree Plot

From Figure 2.15, one can infer the relevance of the attributes. The scree plot indicates that the first attribute is more important than all other attributes.

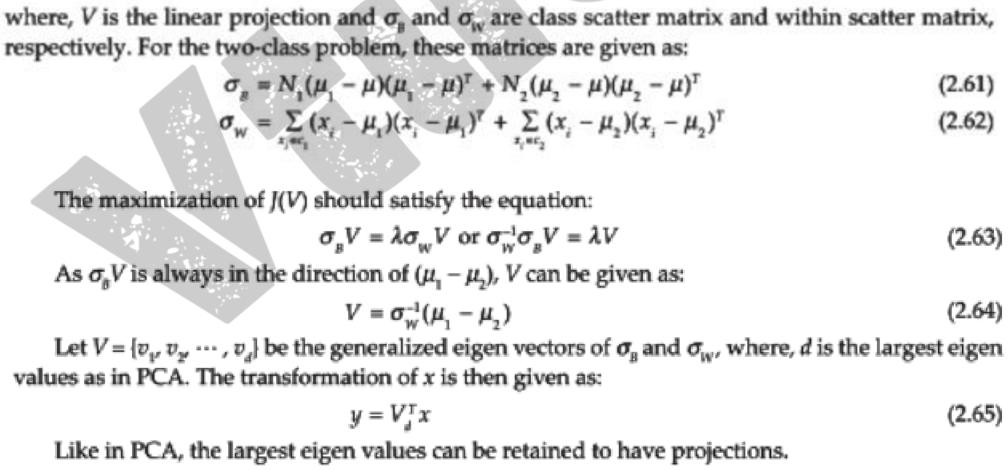
2.10.4 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is also a feature reduction technique like PCA. The focus of LDA is to project higher dimension data to a line (lower dimension data). LDA is also used to classify the data. Let there be two classes, c1 and c2. Let m1 and m2 be the mean of the patterns of two
classes. The mean of the class c1 and c2 can be computed as:

$$\mu_1 = \frac{1}{N_1}\sum_{x_i \in c_1}^{n} x_i \text{ and } \mu_2 = \frac{1}{N_2}\sum_{x_i \in c_2}^{n} x_i$$

The aim of LDA is to optimize the function:

$$J(V) = \frac{V^T \sigma_B V}{V^T \sigma_W V} \qquad (2.60)$$



where, V is the linear projection and σ_B and σ_W are class scatter matrix and within scatter matrix, respectively. For the two-class problem, these matrices are given as:

$$\sigma_B = N_1(\mu_1 - \mu)(\mu_1 - \mu)^T + N_2(\mu_2 - \mu)(\mu_2 - \mu)^T \qquad (2.61)$$

$$\sigma_W = \sum_{x_i \in c_1}(x_i - \mu_1)(x_i - \mu_1)^T + \sum_{x_i \in c_2}(x_i - \mu_2)(x_i - \mu_2)^T \qquad (2.62)$$

The maximization of $J(V)$ should satisfy the equation:

$$\sigma_B V = \lambda \sigma_W V \text{ or } \sigma_W^{-1}\sigma_B V = \lambda V \qquad (2.63)$$

As $\sigma_B V$ is always in the direction of $(\mu_1 - \mu_2)$, V can be given as:

$$V = \sigma_W^{-1}(\mu_1 - \mu_2) \qquad (2.64)$$

Let $V = \{v_1, v_2, \cdots, v_d\}$ be the generalized eigen vectors of σ_B and σ_W, where, d is the largest eigen values as in PCA. The transformation of x is then given as:

$$y = V_d^T x \qquad (2.65)$$

Like in PCA, the largest eigen values can be retained to have projections.

2.10.5 Singular Value Decomposition

Singular Value Decomposition (SVD) is another useful decomposition technique. Let A be the matrix, then the matrix A can be decomposed as:

$$A = USV^T \qquad (2.66)$$

Here, A is the given matrix of dimension m × n, U is the orthogonal matrix whose dimension is m × n, S is the diagonal matrix of dimension n × n, and V is the orthogonal matrix. The procedure for finding decomposition matrix is given as follows:

1.For a given matrix, find AA^T

2.Find eigen values of AA^T
3.Sort the eigen values in a descending order. Pack the eigen vectors as a matrix U.
4.Arrange the square root of the eigen values in diagonal. This matrix is diagonal matrix, S.
5.Find eigen values and eigen vectors for A^TA. Find the eigen value and pack the eigen vector as a matrix called V.

Thus, A = USV^T. Here, U and V are orthogonal matrices. The columns of U and V are left and right singular values, respectively. SVD is useful in compression, as one can decide to retain only a certain component instead of the original matrix A as:

$$a_{ij} = \sum_{k=1}^{H} u_{ik} s_k v_{jk} \tag{2.67}$$

Based on the choice of retention, the compression can be controlled.