Chapter 11

Other Types of Learning

OBJECTIVE OF THE CHAPTER :

In the previous chapters, you have got good understanding of supervised learning algorithm and unsupervised learning algorithm. In this chapter, we will discuss learning types which were not covered in the earlier chapters which includes representation learning, active learning, instance-based Learning, association rule, and ensemble learning. By the end of this chapter, you will gain sufficient knowledge in all the aspects of learning and become ready to start solving problems on your own.

11.1 INTRODUCTION

In this chapter, we will discuss learning types which were not covered in the earlier chapters which include representation learning, active learning, instance-based Learning, association rule, and ensemble learning.

11.2 REPRESENTATION LEARNING

Another name for representation learning is feature learning. While many enterprises nowadays retain substantial amounts of business-related data, most of those are often unstructured and unlabelled. Creating new labels is generally a time-consuming and expensive endeavour. As a result, machine learning algorithms that can straight away mine structures from unlabelled data to improve the business performance are reasonably valued. Representation learning is one such type where the extraction from the overall unlabelled data happens using a 'neural network'. The main objective of representation learning (feature learning) is to find an appropriate representation of data based on features to perform a machine learning task. Representation learning has become a field in itself because of its popularity. Refer the chapter 4 titled ‘Basics of Feature Engineering’ to understand more on feature and feature selection process. An example of some of the characteristics of a triangle are its corners (or vertices), sides, and degree of angle. Every triangle has three corners and three angles. Sum of the three angles of a triangle is equal to 180°. The types of triangles with their characteristics are given below.

Consider you are tasked to classify/identify the types of triangles (Refer **Table 11.1**). The way you do that is to find unique characteristics of the type of triangle given as input.

The system may work something like this

Input – Triangular image

Representation – Three angles, length of sides.

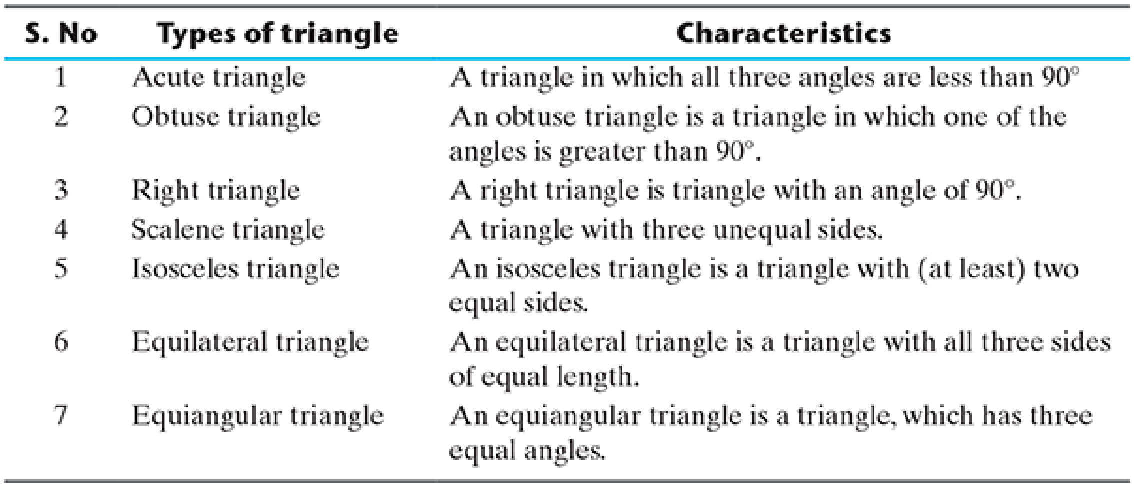
Model – It gets an input representation (angles, length of sides) and applies rules as per Table 11.1 to detect the type of triangle.

Great! Now, we have a primary representational working system to detect the type of triangle.

What happens when we begin to input shapes like square, rectangle, circle, or all sort of shapes instead of a triangle? What if the image contains both a triangle and a circle? Designing to adapt to this kind of features requires deep domain expertise as we start working with real-world applications. Deep Learning goes one step further and learns/tries to learn features on its own. All we would do is to feed in an image as input and let the system learn features like human beings do.

Representation learnings are most widely used in words, speech recognition, signal processing, music, and image identification. For example, a word can have meaning based on the context.

Table 11.1 *Types of Triangle*



S. No	Types of triangle	Characteristics
1	Acute triangle	A triangle in which all three angles are less than 90°
2	Obtuse triangle	An obtuse triangle is a triangle in which one of the angles is greater than 90°.
3	Right triangle	A right triangle is triangle with an angle of 90°.
4	Scalene triangle	A triangle with three unequal sides.
5	Isosceles triangle	An isosceles triangle is a triangle with (at least) two equal sides.
6	Equilateral triangle	An equilateral triangle is a triangle with all three sides of equal length.
7	Equiangular triangle	An equiangular triangle is a triangle, which has three equal angles.

Generation and Recognition in Representation Learning

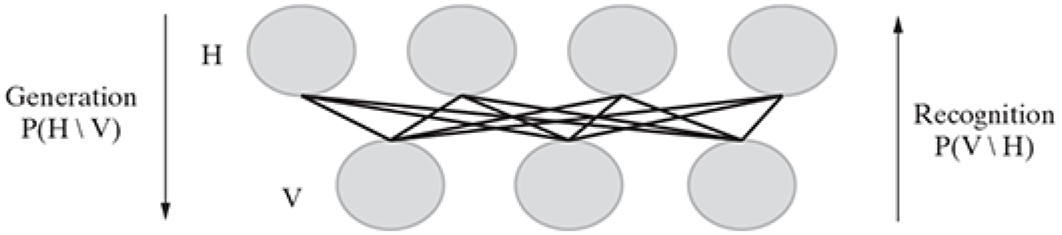


FIG. 11.1 Generation versus recognition

In Figure 11.1, V represents the input data, and H represents the causes. When the causes (H) explains the data (V) we observe, it is called as Recognition. When unknown causes (H) are combined, it can also generate the data (V), it is called as generative (or) Generation.

Representation learning is inspired by the fact that machine learning tasks such as classification regularly requires numerical inputs that are mathematically and computationally easy to process. However, it is difficult to define the features algorithmically for real-world objects such as images and videos. An alternative is to discover such features or representations just through examination, without trusting on explicit algorithms. So, representation learning can be either supervised or unsupervised.

1. Supervised neural networks and multilayer perceptron
2. Independent component analysis (unsupervised)
3. Autoencoders (unsupervised) and
4. Various forms of clustering.

11.2.1 Supervised neural networks and multilayer perceptron

Refer the chapter 10 titled 'Basics of Neural Network' to understand more on the topic of the neural network.

11.2.2 Independent component analysis (Unsupervised)

Independent component analysis, or ICA is similar to principal component analysis (PCA), with the exception that the components are independent and uncorrelated. ICA is used primarily for separating unknown source signals from their observed linear mixtures after the linear mixture with an unfamiliar matrix (A) . Nil information is known about the sources or the mixing process except that there are *N* different recorded mixtures. The job is to recuperate a version(U) of the sources (S) which are identical except for scaling and permutation, by finding a square matrix (W) specifying spatial filters that linearly invert the mixing process, i.e. U = WX. Maximizing the joint entropy, H(y), of the output of a neural processor minimizes the mutual information among the output components (Refer Fig. 11.2).

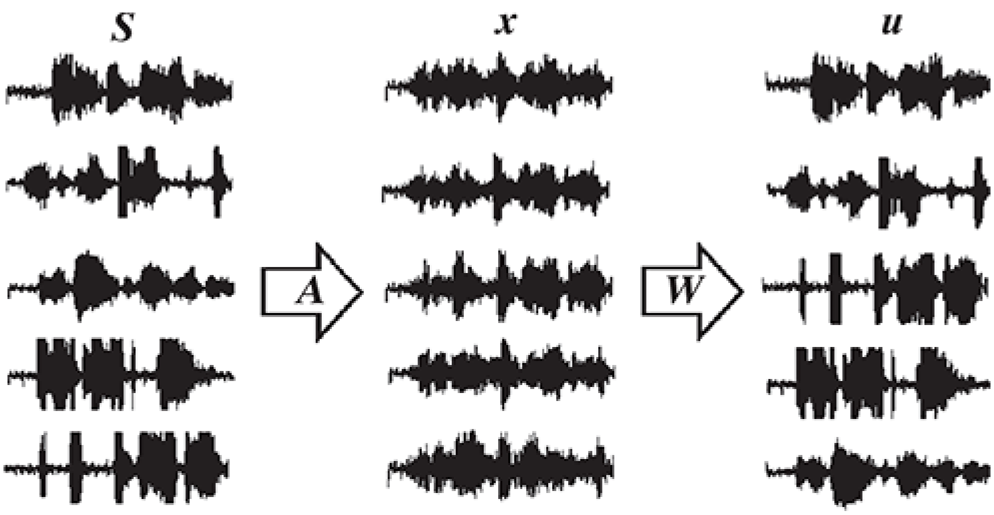


FIG. 11.2 Independent component analysis

11.2.3 Autoencoders

Autoencoders belong to the neural network family (Refer Fig. 11.3) and are similar to Principal Component Analysis (PCA). They are more flexible than PCA. Autoencoders represents both linear and non-linear transformation, whereas PCA represents only linear transformation. The neural network's (autoencoder) target output is its input (x) in a different form (x'). In Autoencoders, the dimensionality of the input is equal to the dimensionality of the output, and essentially what we want is x' = x. x' = Decode (Encode(x))

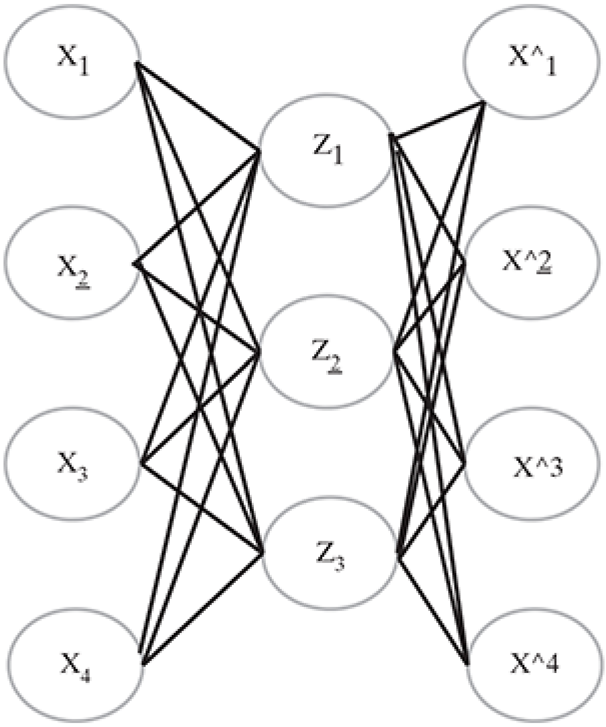


FIG. 11.3 Autoencoders

To get the value from the hidden layer, we multiply the (input to hidden) weights by the input. To get the value from the output, we multiply (hidden to output) weights by the hidden layer values.

Z = f (Wx)

Y = g (Vz)

So Y = g(V f (Wx)) = VWx

So Objective Function J = ∑ [*Xn – VWx (n)*]^2

Heuristics for Autoencoders

Autoencoder training algorithm can be summarized as below:

For each input *x*

1. Do a feed-forward pass to compute activations at all hidden layers, then at the output layer to obtain an output X'

2. Measure the deviation X' from the input X
3. Backpropagate the error through the net and perform weight updates.

11.2.4 Various forms of clustering

Refer chapter 9 titled ‘Unsupervised Learning’ to know more about various forms of Clustering.

11.3 ACTIVE LEARNING

Active learning is a type of semi-supervised learning in which a learning algorithm can interactively query (question) the user to obtain the desired outputs at new data points.

Let P be the population set of all data under consideration. For example, all male students studying in a college are known to have a particularly exciting study pattern.

During each iteration, I, P (total population) is broken up into three subsets.

1. P(K, i): Data points where the label is known (K).
2. P(U, i): Data points where the label is unknown (U)
3. P(C, i): A subset of P(U, i)that is chosen (C) to be labelled.

Current research in this type of active learning concentrates on identifying the best method P(C, i) to chose (C) the data points.

11.3.1 Heuristics for active learning

1. Start with a pool of unlabelled data P(U, i)
2. Pick a few points at random and get their labels P(C, i)
3. Repeat

Some active learning algorithms are built upon aupport vector machines (SVMs) to determine the data points to label. Such methods usually calculate the margin (W) of each unlabelled datum in P(U, i) and assume W as an *n*-dimensional space distance commencing covering datum and the splitting hyperplane.

Minimum marginal hyperplane (MMH) techniques consent that the data records with the minimum margin (W) are those that the SVM is most uncertain about and therefore should be retained in P(C, i) to be labelled. Other comparable approaches, such as Maximum Marginal Hyperplane, select data with the significant W (*n*-dimensional space). Other approaches to select a combination of the smallest as well as largest Ws.

11.3.2 Active learning query strategies

Few logics for determining which data points should be labelled are

1. Uncertainty sampling
2. Query by committee
3. Expected model change
4. Expected error reduction
5. Variance reduction

Uncertainty sampling: In this method, the active learning algorithm first tries to label the points for which the current model is least specific (as this means a lot of confusion) on the correct output.

Query by committee: A range of models are trained on the current labelled data, and vote on the output for unlabelled data; label those points for which the 'committee' disagrees the most.

Expected model change: In this method, the active learning algorithm first tries to label the points that would most change the existing model itself.

Expected error reduction: In this method, the active learning algorithm first tries to label the points that would mostly reduce the model's generalization error.

Variance reduction: In this method, the active learning algorithm first tries to label the points that would reduce output variance (which is also one of the components of error).

11.4 INSTANCE-BASED LEARNING (MEMORY-BASED LEARNING)

In instance-based learning (memory-based learning), the algorithm compares new problem instances with instances already seen in training, which have been stored in memory for reference rather than performing explicit generalization. It is called instance-based as it constructs hypotheses from the training instances (memories) themselves. Computational complexity O(n) of the hypothesis can grow when we have some data (*n*). It can quickly adapt its model to previously unseen data. New instances are either stored or thrown away based on the previously set criteria.

Examples of instance-based learning include *K*-nearest neighbour algorithm (*K*-NN), kernel machines (support vector machine, PCA), and radial basis function (RBF) networks.

All these algorithms store already known instances and compute distances or similarities between this instance and the training instances to make the final decision. Instance reduction algorithm can be used to overcome memory complexity problem and overfitting problems. Let us look into radial basis function (RBF) in detail.

11.4.1 Radial basis function

Radial basis function (RBF) networks typically have three layers: one input layer, one hidden layer with a non-linear RBF activation function, and one linear output layer (Refer Fig. 11.4).

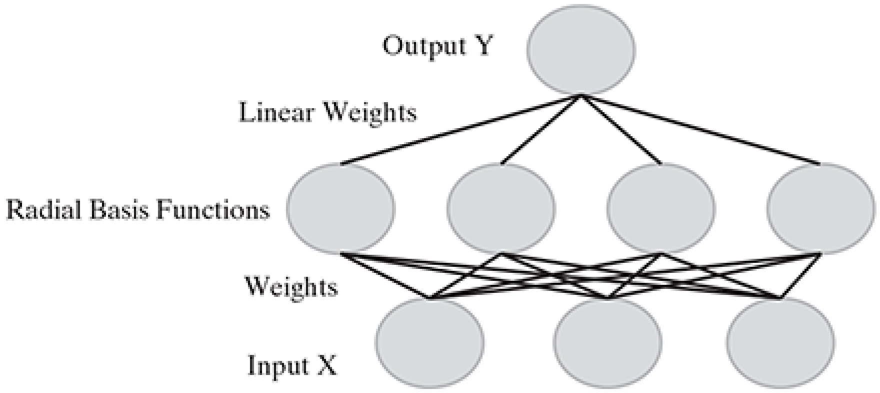


FIG. 11.4 Radial Basis Function

In the above diagram, an input vector(x) is used as input to all radial basis functions, each with different parameters. The input layer (X) is merely a fan-out layer, and no processing happens here. The second layer (hidden) performs radial basis functions, which are the non-linear mapping from the input space (Vector X) into a higher order dimensional space. The output of the network (Y) is a linear combination of the outputs from radial basis functions. If pattern classification is required (in Y), then a hard-limiter or sigmoid function could be placed on the output neurons to give 0/1 output values.

The distinctive part of the radial basis function network (RFFN) is the procedure implemented in the hidden layer. The idea is that the patterns in the input space form clusters. Beyond this area (clustering area, RFB function area), the value drops dramatically. The Gaussian function is the most commonly used radial-basis function. In an RBF network, *r* is the distance from the cluster centre.

Space (distance) computed from the cluster centre is usually the Euclidean distance. For each neuron that is part of the hidden layer, the weights represent the coordinates of the centre of the cluster. Therefore, when that neuron receives an input pattern, *X*, the distance is found using the following equation:

$$r_j = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2}$$

The variable sigma, σ, denotes the width or radius of the bell-shape and is to be determined by calculation. When the distance from the centre of the Gaussian reaches σ, the output drops from 1 to 0.6.

$$(hidden_{unit})\phi_j = \exp\left(-\frac{\sum_{i=1}^{n} (x_i - w_{ij})^2}{2\sigma^2}\right)$$

The output $y(t)$ is a weighted sum of the outputs of the hidden layer, given by

$$y(t) = \sum_{i=1}^{n} w_i \phi(\| u(t) - c_i \|),$$

where

$u(t)$ is the input

$\phi(.)$ is an arbitrary non-linear radial basis function

$\|.\|$ denotes the norm that is usually assumed to be Euclidean

c_i are the known centres of the radial basis functions

w_i are the weights of the function

In radial functions, the response decreases (or increases) monotonically with distance from a central point and they are radially symmetric. The centre, the distance scale, and the exact shape of the radial function are the necessary considerations of the defined model. The most commonly used radial function is the Gaussian radial filter, which in case of a scalar input is

$$h(x) = \exp\left(-\frac{(x-c)^2}{\beta^2}\right)$$

Its parameters are its centre c and its radius β (width), illustrates a Gaussian RBF with centre c = 0 and radius β = 1. A Gaussian RBF monotonically decreases with distance from the centre.

11.4.2 Pros and cons of instance-based learning method

Advantage: Instance-based learning (IBL) methods are particularly well suited to problems where the target function is highly complex, but can also be defined by a group of not as much of complex local approximations.

The disadvantage I: The cost of classification of new instances can be high (a majority of the calculation takes place at this stage).

Disadvantage II: Many IBL approaches usually study all characteristics of the instances leading to dimensionality-related problems.

11.5 ASSOCIATION RULE LEARNING ALGORITHM

Association rule learning is a method that extracts rules that best explain observed relationships between variables in data. These rules can discover important and commercially useful associations in large multidimensional data sets that can be exploited by an organization. The most popular association rule learning algorithms are Apriori algorithm and Eclat process.

11.5.1 Apriori algorithm

Apriori is designed to function on a large database containing various types of transactions (for example, collections of products bought by customers, or details of websites visited by customers frequently). Apriori algorithm uses a ‘bottom-up’ method, where repeated subsets are extended one item at a time (a step is known as candidate generation, and groups of candidates are tested against the data). The Apriori Algorithm is a powerful algorithm for frequent mining of itemsets for boolean association rules. Refer chapter 9 titled 'unsupervised learning' to understand more on this algorithm.

11.5.2 Eclat algorithm

ECLAT stands for '**E**quivalence Class **C**lustering and bottom-up **Lat**tice Traversal'. Eclat algorithm is another set of frequent itemset generation similar to Apriori algorithm. Three traversal approaches such as 'Top-down', 'Bottom-up', and Hybrid approaches are supported. Transaction IDs (TID list) are stored here. It represents the data in vertical format.

Step 1: Get Transaciton IDs : tidlist for each item (DB scan).

Step 2. Transaciton IDs (tidlist) of {a} is exactly the list of transactions covering {a}.

Step 3. Intersect tidlist of {a} with the tidlists of all other items, resulting in tidlists of {a,b}, {a,c}, {a,d}, … = {a}–conditional database (if {a} removed).

Step 4. Repeat from 1 on {a}–conditional database 5. Repeat for all other items.

11.6 ENSEMBLE LEARNING ALGORITHM

Ensemble means collaboration/joint/group. Ensemble methods are models that contain many weaker models that are autonomously trained and whose forecasts are combined approximately to create the overall prediction model. More efforts are required to study the types of weak learners and various ways to combine them.

The most popular ensemble learning algorithms are

Bootstrap Aggregation (bagging)

Boosting

AdaBoost

Stacked Generalization (blending)

Gradient Boosting Machines (GBM)

Gradient Boosted Regression Trees (GBRT)

Random Forest

11.6.1 Bootstrap aggregation (Bagging)

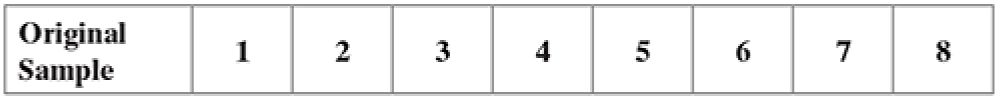
Bootstrap Aggregation (bagging) works using the principle of the Bootstrap sampling method. Given a training data set D containing m examples, bootstrap drawing method draws a sample of training examples $D_{\boldsymbol{i}}$, by selecting m examples in uniform random with replacement. It comprises two phases namely Training phase and Classification Phase.

Training Phase:

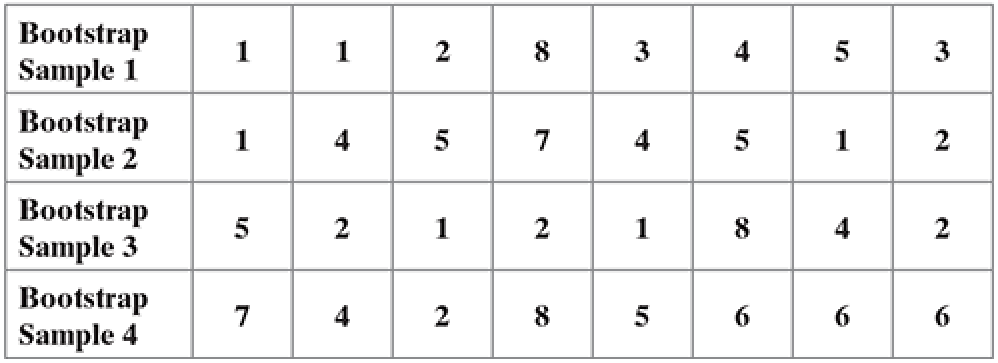
1. Initialize the parameters
2. D = {Φ }
3. H = the number of classification
4. For k = 1 to h
5. Take a bootstrap sample $S_{\boldsymbol{k}}$ from training set S
6. Build the classifier $D_{\boldsymbol{k}}$ using $S_{\boldsymbol{k}}$ as a training set
7. $D = D \cup D_{\boldsymbol{i}}$
8. Return D

Classification Phase:

1. Run $D_{\boldsymbol{1}}$, $D_{\boldsymbol{2}}$,..........$D_{\boldsymbol{k}}$ on the input k
2. The class with a maximum number of the vote is chosen as the label for X.



Original Sample	1	2	3	4	5	6	7	8



Bootstrap Sample 1	1	1	2	8	3	4	5	3
Bootstrap Sample 2	1	4	5	7	4	5	1	2
Bootstrap Sample 3	5	2	1	2	1	8	4	2
Bootstrap Sample 4	7	4	2	8	5	6	6	6

In the above example, the original sample has eight data points. All the Bootstrap samples also have eight data points. Within the same bootstrap sample, data points are repeated due to selection with the replacement.

11.6.2 Boosting

Just like bagging, boosting is another key ensemble-based technique. Boosting is an iterative technique. It decreases the biasing error. If an observation was classified incorrectly, then it boosts the weight of that observation. In this type of ensemble, weaker learning models are trained on resampled data, and the outcomes are combined using a weighted voting approach based on the performance of different models. Adaptive boosting or AdaBoost is a special variant of a boosting algorithm. It is based on the idea of generating weak learners and learning slowly.

11.6.3 Gradient boosting machines (GBM)

As we know already, boosting is the machine learning algorithm, which boosts week learner to strong learner. We may already know that a learner estimates a target function from training data. Gradient boosting is the boosting with Gradient.

Gradient: Consider L as a function of an ith variable, other $N - 1$ variables are fixed at the current point, by calculating derivative of L at that point (L is differentiable), we have – if the derivative is positive/negative –

a decrease/increase value of the *i*th variable in order to make the loss smaller (*). Applying the above calculation for all *i* = 1… *N*, we will get *N* derivative values forming a vector, so-called gradient of an *N*-variable function at the current point.

11.7 REGULARIZATION ALGORITHM

This is an extension made to another method (typically regression method) that penalizes models based on their complexity, favouring simpler models that are also better at generalizing. Regularization algorithms have been listed separately here because they are famous, influential, and generally simple modifications made to other methods.

The most popular regularization algorithms are

Ridge Regression

Least Absolute Shrinkage and Selection Operator (LASSO)

Elastic Net

Least-Angle Regression (LARS)

Refer chapter 8 titled “Supervised Learning: Regression” for Ridge regression and the LASSO method.

Elastic Net

When we are working with high-dimensional data sets with a large number of independent variables, correlations (relationships) amid the variables can be often result in multicollinearity. These correlated variables which are strictly related can sometimes form groups or clusters called as an elastic net of correlated variables. We would want to include the complete group in the model selection even if just one variable has been selected.

11.8 SUMMARY

- Another name for Representation learning is feature learning.
- Representation learnings are most widely used in words, speech recognition, signal processing, music, and images identification. For example, a word can have meaning based on the context.
- Independent component analysis, or ICA) is similar to principal components analysis (PCA), with the exception that the components are independent and uncorrelated.
- Autoencoders belongs to neural network family and are similar to PCA (Principal Component Analysis). They are more flexible than PCA.
- Active learning is a type of semi-supervised learning in which a learning algorithm can interactively query (question) the user to obtain the desired outputs at new data points.
- Uncertainty sampling: In this method, the active learning algorithm first tries to label the points for which the current model is least specific (a that means much confusion) on the correct output.
- In instance-based learning (memory-based learning), the algorithm compares new problem instances with instances already seen in training, which have been stored in memory for reference rather than performing explicit generalization.
- Examples of instance-based learning include K-nearest neighbor algorithm (K-NN), Kernel machines (Support Vector Machine, PCA) and Radial Basis Function networks (RBF networks).
- Radial basis function (RBF) networks typically have three layers: one input layer, one hidden layer with a non-linear RBF activation function and one linear output layer.
- Association rule learning is methods that extract rules that best explain observed relationships between variables in data. These rules can discover important and commercially useful associations in large multidimensional datasets that can be exploited by an organization.
- Apriori is designed to function on a large database containing various types of transactions (for example, collections of products bought by customers, or details of websites visited by customers frequently).

- ECLAT stands for “Equivalence Class Clustering and bottom-up Lattice Traversal.” Eclat algorithm is another set of frequent itemset generation similar to Apriori algorithm. Three traversal approaches such as “Top-down,” “Bottom-up” and Hybrid approaches are supported.
- Ensemble means collaboration/joint/Group. Ensemble methods are models that contain many weaker models that are autonomously trained and whose forecasts are joined approximately to create the overall prediction model.

SAMPLE QUESTIONS

MULTIPLE-CHOICE QUESTIONS (1 MARK QUESTIONS)

1. Another name for Representation learning is ___
    1. Feature learning
    2. Active learning
    3. Instance-based learning
    4. Ensemble learnings
2. In representation learning, when unknown causes (H) are combined, it can also generate the data (V), it is called as ___
    1. Recognition
    2. Generation
    3. Representative
    4. Combinative
3. When the causes (H) explains the data (V) we observe, it is called as ___
    1. Recognition
    2. Generation
    3. Representative
    4. Combinative
4. When unknown causes (H) are combined, it can also generate the data (V), it is called as ___
    1. Recognition
    2. Generation
    3. Representative
    4. Combinative

5. Independent component analysis is similar to
   1. Dependent analysis
   2. Sub Component analysis
   3. Data analysis
   4. Principal component analysis
6. In this method, the active learning algorithm first tries to label the points for which the current model is least specific (as this means a lot of confusion) on the correct output.
   1. Expected model change
   2. Query by committee
   3. Uncertainty sampling
   4. Expected error reduction
7. A range of models are trained on the current labeled data, and vote on the output for unlabeled data; label those points for which the 'Group of people' disagrees the most.
   1. Expected model change
   2. Query by committee
   3. Uncertainty sampling
   4. Expected error reduction
8. In this method, the active learning algorithm first tries to label the points that would most change the existing model itself
   1. Expected model change
   2. Query by committee
   3. Uncertainty sampling
   4. Expected error reduction
9. In this method, the active learning algorithm first tries to label the points that would mostly reduce the model's generalization problem.
   1. Expected model change
   2. Query by committee
   3. Uncertainty sampling
   4. Expected error reduction
10. Which of the following is not a type of Ensemble learning algorithms?
    1. Boosting
    2. AdaBoost
    3. GBM

4. Elastic Net

SHORT-ANSWER TYPE QUESTIONS (5 MARKS QUESTIONS)

1. What is the main objective of Representation Learning?
2. What are the applications of Representational Learning?
3. Define Independent Component Analysis? What is the primary usage of it?
4. What is Uncertainty sampling in Active learning?
5. What is Query by committee in Active learning?
6. What are the Pros and Cons of Instance-Based Learning (IBL) Method?
7. Write notes on Elastic Net
8. List down various types of regularization algorithms
9. What is Gradient?
10. What is Bootstraped Aggregation (Bagging)?
11. List down various types of Ensemble learning algorithms

LONG-ANSWER TYPE QUESTIONS (10 MARKS QUESTIONS)

1. Derive primary representational working system to detect the type of triangle.
2. Discuss Generation and Recognition in Representation Learning
3. Discuss Independent component analysis (unsupervised)
4. Discuss Autoencoders in detail with diagram
5. What is active learning? Explain its heuristics
6. Discuss various Active learning Query Strategies
7. Discuss Instance-based Learning (Memory-based learning)
8. Discuss Radial Basis Function in detail
9. Discuss various Association Rule Learning Algorithm in detail
10. What is Ensemble Learning Algorithm? Discuss various types.