**MACHINE LEARNING – WORKSHEET 3**

**SOLUTIONS**

**1. Give short description each of Linear, RBF, Polynomial kernels used in SVM.**

**Ans:** Kernels in SVM classification refer to the function that is responsible for defining the decision boundaries between the classes. Different types of kernels are as follows

* **Linear Kernel** :- It is used when the data is Linearly separable, that is, it can be separated using a straight line. It is one of the most common kernels to be used. It is mostly used when there are a Large number of Features in a particular Data Set. One of the examples where there are a lot of features, is Text Classification, as each alphabet is a new feature. So we mostly use Linear Kernel in Text Classification.
* **Radial basis function kernel (RBF)/ Gaussian Kernel:** Radial Basis Function is another popular Kernel method used in SVM models for more. RBF kernel is a function whose value depends on the distance from the origin or from some point. RBF kernel is used when the boundaries are hypothesized to be curve-shaped.
* RBF kernel uses two main parameters, gamma and C that are related to:

1. the decision region (how spread the region is), and

2. the penalty for misclassifying a data point respectively.

* **Polynomial kernel:-**  represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.Intuitively, the polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In the polynomial kernel, we simply calculate the dot product by increasing the power of the kernel.

|  |  |
| --- | --- |
| **S.No** | **Main Differences** |
| 1 | **time of SVM learning:** linear < poly < rbf |
| 2 | **ability to fit any data:** linear < poly < rbf |
| 3 | **risk of overfitting:** linear < poly < rbf |
| 4 | **number of hyperparameters:** linear (0) < rbf (2) < poly (3) |
| 5 | **how "local" is particular kernel:** linear < poly < rbf |
| 6 | **risk of underfitting:** rbf < poly < linear |

**2. R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of goodness of fit of model in regression and why??**

**Ans: RSS** (**Residual Sum of Squares**) is a better measure of goodness of fit of model in regression because R-squared can measure the goodness of out-of-sample prediction. If we split our data into training and testing parts and fit a regression model on the training one, we can get a valid R squared value on training part, but we can't legitimately compute an R squared on the test part.

**3.** **What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sum of Squares) in regression. Also mention the equation relating these three metrics with each other.**

**Ans:**

**Explained sum of square (ESS)** or **Regression sum of squares** or Model sum of squares is a statistical quantity used in modelling of a process. ESS gives an estimate of how well a model explains the observed data for the process.

It tells how much of the variation between observed data and predicted data is being explained by the model proposed. Mathematically, it is the sum of the squares of the difference between the predicted data and mean data.

image2.png

**RSS (Residual Sum of Squares):**

A residual sum of squares (RSS) is a statistical technique used to measure the amount of [variance](https://www.investopedia.com/terms/v/variance.asp) in a data set that is not explained by a regression model. This expression is also known as unexplained variation and is the portion of total variation that measures discrepancies (errors) between the actual values of Y and those estimated by the regression equation.

image3.png

**TSS (Total Sum of Squares):**

Total sum of squares (TSS) is a quantity that appears as part of a standard way of presenting results of such analyses. It is defined as being the sum, over all observations, of the squared differences of each observation from the overall mean. In statistical linear models, (particularly in standard regression models), the TSS is the sum of the squares of the difference of the dependent variable and its grand mean.

**TSS (Total Sum of Squares)= Explained sum of square (ESS)+ RSS (Residual Sum of Squares)**

**4. What is Gini –impurity index?**

**Ans:**

* Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. Here **impurity** means, if all the elements belong to a single class, then it can be called pure.
* The degree of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes. A Gini Index of 0.5 denotes equally distributed elements into some classes.

**5. Are unregularized decision-trees prone to overfitting? If yes, why?**

**Ans:**  Yes, un-regularized decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions. Unregularized tress grow to a depth until they don’t get pure leaf nodes. Sometimes, it may happen that there is only one point in each leaf nodes. Now, changing a single point in dataset will cause a change in the entire model, which is nothing but **over-fitting** but as always, this will be at the expense of error due to bias.

Random forests mitigate this problem well.

**6. What is an ensemble technique in machine learning?**

**Ans:**

* **Ensemble methods** is a machine learning technique that combines **several base models** in order to produce **one optimal predictive** model.
* Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would.

Types of Ensemble Techniques:

## 1. Bootstrap Aggregating (also known as “Bagging”)

## 2. Boosting: Converting Weak Models to Strong Ones

**7. What is the difference between Bagging and Boosting techniques?**

**Ans:**

**Bagging:** Bagging is used when the goal is to reduce the variance of a decision tree classifier**. Here the objective is to create several subsets of data from training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees.** As a result, we get an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree classifier.

**Boosting**: Boosting is used to create a collection of predictors. **In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analysing data for errors. Consecutive trees (random sample) are fit and at every step, the goal is to improve the accuracy from the prior tree.**When an input is misclassified by a hypothesis, its weight is increased so that next hypothesis is more likely to classify it correctly. This process converts weak learners into better performing model.

**Basic difference between Bagging and Boosting:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Bagging** | **Boosting** |  |
| **Partitioning of data** | Random | Higher vote to misclassified samples |  |
| **Goal to achieve** | Minimum variance | Increase accuracy |  |
| **Methods used** | Random subspace | Gradient descent |  |
| **Functions to combine single model** | Weighted average | Weighted majority vote |  |
| **Example** | Random Forest | Ada Boost |  |
| **Overfitting** | Reduces over-fitting of  the model. | Prone to over-fitting. |  |

**8. What is out-of-bag error in random forests?**

**ANS:** Out-of-bag error, also called out-of-bag estimate, is a method of measuring the prediction error of random forests, boosted decision trees, and other machine learning models utilizing bootstrap aggregating to sub-sample data samples used for training.

**9. What is K-fold cross-validation?**

**Ans: Cross Validation** is a very useful technique for assessing the performance of machine learning models. It helps in knowing how the machine learning model would generalize to an independent data set.

**K-fold cross-validation:**

* In this method, the training dataset will be split into multiple ‘k’ smaller parts/sets. Hence the name ‘k’-fold.
* The current training dataset would now be divided into ‘k’ parts, out of which one dataset is left out and the remaining ‘k-1’ datasets are used to train the model.
* This is done multiple number of times. The number of times that it has to be done is mentioned by the user in the code.
* The one that was kept out of the training is used as a ‘validation dataset’. This can be used to tune hyperparameters and see how the model performs and change the values accordingly, to yield better results.
* Even though the size of the dataset isn’t reduced considerably, it was reduced to a certain extent. This method also makes sure that the model remains robust and generalizes well on the data.

**10. What is hyper parameter tuning in machine learning and why it is done?**

**Ans:**

* Hyperparameter tuning is the process of tuning the parameters present as the tuples while we build machine learning models. Machine learning algorithms never learn these parameters. These are tuned so that we could get good performance by the model.
* Hyperparameter tuning aims to find such parameters where the performance of the model is highest or where the **model performance is best** and the error rate is least.

Two best strategies for Hyperparameter tuning are:

1. [GridSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)
2. [RandomizedSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html)

**11. What issues can occur if we have a large learning rate in Gradient Descent?**

**Ans:**

* When the learning rate is too large, gradient descent can **inadvertently increase** rather than decrease the training error.
* A large learning rate allows the model to learn faster**, at the cost of arriving** on a sub-optimal final set of weights.
* A learning rate that is too large will result in weight updates that will be too large and the performance of the model (such as its loss on the training dataset) will oscillate over training epochs. Oscillating performance is said to be caused by weights that diverge.
* We may get overshoot and land on the other side of minima, for this reason we have to come back to minima. In this approach if our learning rate is large we will keep on moving or jumping around our minima.

**12. What is bias-variance trade off in machine learning?**

**Ans:**

* The **bias-variance trade-off** is a central problem in supervised learning. One wants to [choose a model](https://en.wikipedia.org/wiki/Model_selection) that both accurately captures the regularities in its training data, but also [generalizes](https://en.wikipedia.org/wiki/Generalization) well to unseen data. Unfortunately, it is typically impossible to do both simultaneously.
* The bias–variance dilemma or bias–variance problem is the conflict in which models with a lower [bias](https://en.wikipedia.org/wiki/Bias_of_an_estimator) have a higher [variance](https://en.wikipedia.org/wiki/Variance) of the parameter estimates across [samples](https://en.wikipedia.org/wiki/Sample_(statistics)), and vice versa. It’s more of like inverse proportionality between them.
* The [bias error](https://en.wikipedia.org/wiki/Bias_of_an_estimator) leads to underfitting while the variance leads to overfitting thus this bias-variance tradeoff occurs and one have to compromise with atleast one of both.

**13. What is the need of regularization in machine learning?**

**Ans:** Regularization is a techniqueused in an attempt to solve the following problems:

* Overfitting: Overfitting which results in model failing to generalize on the unseen dataset.
* Multicollinearity: Model suffering from multicollinearity effect.
* Computationally Intensive: A model becomes computationally intensive.

Thus in order to remove these problems regularization should be done using following techniques:

* Ridge regression (**L2 norm**)
* Lasso regression (**L1 norm**)
* Elastic net regression

**14. Differentiate between Adaboost and Gradient Boosting**

**Ans:**

|  |  |
| --- | --- |
| **Gradient boosting** | **Ada Boosting** |
| This approach trains learners based upon minimising the loss function of a learner (i.e., training on the residuals of the model | This method focuses on training upon misclassified observations. Alters the distribution of the training dataset to increase weights on sample observations that are difficult to classify. |
| Weak learners are decision trees constructed in a greedy manner with split points based on purity scores (i.e., Gini, minimise loss). Thus, larger trees can be used with around 4 to 8 levels. Learners should still remain weak and so they should be constrained (i.e., the maximum number of layers, nodes, splits, leaf nodes) | The weak learners in case of adaptive boosting are a very basic form of decision tree known as stumps. |
| All the learners have equal weights in the case of gradient boosting. The weight is usually set as the learning rate which is small in magnitude. | The final prediction is based on a majority vote of the weak learners’ predictions weighted by their individual accuracy. |

**15. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?**

**Ans:** No, we cannot use Logistic Regression for classification of Non-Linear Data because Logistic Regression is known and used as a linear classifier. It is used to come up with a hyperplane in feature space to separate observations that belong to a class from all the other observations that do not belong to that class. The decision boundary is thus linear.