**MACHINE LEARNING – WORKSHEET 3**

**Solution**

**Q1 to Q15 are subjective answer type questions, Answer them briefly.**

**Q1. Give short description each of Linear, RBF, Polynomial kernels used in SVM.**

Ans.

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example*linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.*  
Introduce Kernel functions for sequence data, graphs, text, images, as well as vectors. The most used type of kernel function is RBF. Because it has localized and finite response along the entire x-axis.  
The kernel functions return the inner product between two points in a suitable feature space. Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.

**Examples of SVM Kernels:**

**Polynomial kernel: -** It is popular in image processing.

[Polynomial kernel equation](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/polynomial-kernel.png)

*Polynomial kernel equation*

Where d is the degree of the polynomial.

**Gaussian kernel: -** It is a general-purpose kernel; used when there is no prior knowledge about the data.

[Gaussian kernel equation](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/gaussian-kernel.png)

**Gaussian radial basis function (RBF):-** It is a general-purpose kernel; used when there is no prior knowledge about the data.

[Gaussian radial basis function (RBF)](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/gaussian-radial-basis-function-RBF.png)

[Gaussian radial basis function (RBF)](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/gaussian-radial-basis-function-RBF-1.png)

Sometimes parametrized using:

[Gaussian radial basis function (RBF)](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/gaussian-radial-basis-function-RBF-2.png)

**Laplace RBF kernel: -** It is general-purpose kernel; used when there is no prior knowledge about the data.  
Equation is:

[Laplace RBF kernel equation](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/laplace-RBF-kernel.png)

**Hyperbolic tangent kernel: -** We can use it in neural networks.

[Hyperbolic tangent kernel equation](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/hyperbolic-tangent-kernel.png)

, for some (not every) k>0 and c<0.

**Sigmoid kernel: -** We can use it as the proxy for neural networks. Equation is

[ Sigmoid kernel equation](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2017/08/sigmoid-kernel.png)

**Q2. 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.**

**Residual:** for a point in the data is the difference between the actual value and the value predicted by our linear regression model.

Residual

Residual plots tell us whether the regression model is the right fit for the data or not. It is actually an assumption of the regression model that there is no trend in residual plots

**Residual sum of squares (RSS)**: - Using the residual values, we can determine the sum of squares of the residuals also known as **Residual sum of squares** or RSS.

Residual Sum of Squares

The lower the value of RSS, the better is the model predictions. Or we can say that – a regression line is a line of best fit if it minimizes the RSS value. But there is a flaw in this – RSS is a scale variant statistic. Since RSS is the sum of the squared difference between the actual and predicted value, the value depends on the scale of the target variable.

**R-squared**: - R-squared statistic or coefficient of determination is a scale invariant statistic that gives the proportion of variation in target variable explained by the linear regression model.

It gives the degree of variability in the target variable that is explained by the model or the independent variables. If this value is 0.7, then it means that the independent variables explain 70% of the variation in the target variable.

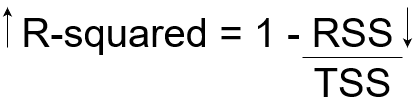
R-squared value always lies between 0 and 1. A higher R-squared value indicates a higher amount of variability being explained by our model and vice-versa.

*R-squared = (TSS-RSS)/TSS*

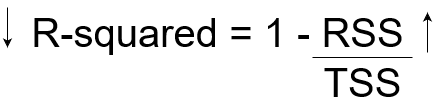
*= Explained variation/ Total variation*

*= 1 – Unexplained variation/ Total variation*

If we had a really low RSS value, it would mean that the regression line was very close to the actual points. This means the independent variables explain the majority of variation in the target variable. In such a case, we would have a really high R-squared value.



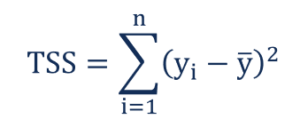
On the contrary, if we had a really high RSS value, it would mean that the regression line was far away from the actual points. Thus, independent variables fail to explain the majority of variation in the target variable. This would give us a really low R-squared value.



**Q3. 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.**

#### **Total sum of squares: -** The total sum of squares is a variation of the values of a [dependent variable](https://corporatefinanceinstitute.com/resources/knowledge/terms/dependent-variable/) from the sample mean of the dependent variable. Essentially, the total sum of squares quantifies the total variation in a [sample](http://www.webmath.com/sampledata.html). It can be determined using the following formula:



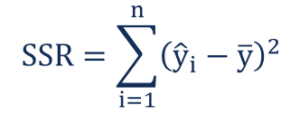
Where:

* yi – the value in a sample
* ȳ – the mean value of a sample

#### **Regression sum of squares: -** (also known as the sum of squares due to regression or explained sum of squares)

The regression sum of squares describes how well a regression model represents the modeled data. A higher regression sum of squares indicates that the model does not fit the data well.

The formula for calculating the regression sum of squares is:



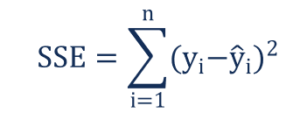
Where:

* ŷi – the value estimated by the regression line
* ȳ – the mean value of a sample

#### **Residual sum of squares**:- (also known as the sum of squared errors of prediction)

The residual sum of squares essentially measures the variation of modeling errors. In other words, it depicts how the variation in the dependent variable in a regression model cannot be explained by the model. Generally, a lower residual sum of squares indicates that the regression model can better explain the data while a higher residual sum of squares indicates that the model poorly explains the data.

The residual sum of squares can be found using the formula below:



Where:

* yi – the observed value
* ŷi – the value estimated by the regression line

The relationship between the three types of sum of squares can be summarized by the following equation:

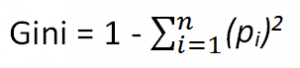
Relationship Formula

**Q4. 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. But what is actually meant by ‘impurity’? 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.

**Formula for Gini Index**



Where *pi is* the probability of an object being classified to a particular class.

While building the [decision tree](https://quantra.quantinsti.com/course/decision-trees-analysis-trading-ernest-chan), we would prefer choosing the attribute/feature with the least Gini index as the root node.

**Q5. Are unregularized decision-trees prone to overfitting? If yes, why?**

**Ans.**

**Q6. 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. To better understand this definition lets take a step back into ultimate goal of machine learning and model building. This is going to make more sense as I dive into specific examples and why Ensemble methods are used.  
**Types of Ensemble Methods**

* ***BAGG*ing, or *B*ootstrap *AGG*regating. BAGGing** gets its name because it combines *B*ootstrapping and *Agg*regation to form one ensemble model. Given a sample of data, multiple bootstrapped subsamples are pulled. A Decision Tree is formed on each of the bootstrapped subsamples. After each subsample Decision Tree has been formed, an algorithm is used to aggregate over the Decision Trees to form the most efficient predictor. The image below will help explain:
* **Random Forest** Models. Random Forest Models can be thought of as **BAGG**ing, with a slight tweak. When deciding where to split and how to make decisions, BAGGed Decision Trees have the full disposal of features to choose from. Therefore, although the bootstrapped samples may be slightly different, the data is largely going to break off at the same features throughout each model. In contrary, Random Forest models decide where to split based on a random selection of features.

**Q7. What is the difference between Bagging and Boosting techniques?**

**Ans.**

|  |  |
| --- | --- |
| **Bagging** | **Boosting** |
| 1. Simplest way of combining predictions that belong to the same type. | A way of combining predictions that belong to the different types. |
| 1. Aim to decrease variance, not bias. | Aim to decrease bias, not variance. |
| 1. Each model receives equal weight. | Models are weighted according to their performance |
| 1. Each model is built independently. | New models are influenced by performance of previously built models. |
| 1. Different training data subsets are randomly drawn with replacement from the entire training dataset. | Every new subsets contains the elements that were misclassified by previous models |
| 1. Bagging tries to solve over-fitting problem. | Boosting tries to reduce bias. |
| 1. If the classifier is unstable (high variance), then apply bagging. | If the classifier is stable and simple (high bias) the apply boosting. |
|  |  |

**Q8. What is out-of-bag error in random forests?**

**Ans.**

**Out-of-bag (OOB**) error, also called out-of-bag estimate, is a method of measuring the prediction error of [random forests](https://en.wikipedia.org/wiki/Random_forest), [boosted decision trees](https://en.wikipedia.org/wiki/Gradient_boosting), and other [machine learning](https://en.wikipedia.org/wiki/Machine_learning) models utilizing [bootstrap aggregating](https://en.wikipedia.org/wiki/Bootstrap_aggregating) (bagging) to sub-sample data samples used for training. OOB is the mean prediction error on each training sample xᵢ, using only the trees that did not have xᵢ in their bootstrap sample.

Subsampling allows one to define an out-of-bag estimate of the prediction performance improvement by evaluating predictions on those observations which were not used in the building of the next base learner.

**Q9. What is K-fold cross-validation?**

**Ans.**

## K-Fold Cross-Validation: - Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

**Q10. What is hyper parameter tuning in machine learning and why it is done?**

**Ans.**

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters.  
However, there is another kind of parameters, known as Hyperparameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Some examples of model hyperparameters include:

1. The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
2. The learning rate for training a neural network.
3. The C and sigma hyperparameters for support vector machines.
4. The k in k-nearest neighbors.

The aim of this article is to explore various strategies to tune hyperparameter for Machine learning model. Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:

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

**GridSearchCV**:- In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values. This approach is called GridSearchCV, because it searches for best set of hyperparameters from a grid of hyperparameters values.

*Drawback* : GridSearchCV will go through all the intermediate combinations of hyperparameters which makes grid search computationally very expensive.  
   
**RandomizedSearchCV:-**   
RandomizedSearchCV solves the drawbacks of GridSearchCV, as it goes through only a fixed number of hyperparameter settings. It moves within the grid in random fashion to find the best set hyperparameters. This approach reduces unnecessary computation.

**Q11. What issues can occur if we have a large learning rate in Gradient Descent?**

**Ans.**

Deep learning neural networks are trained using the stochastic gradient descent algorithm.

**Stochastic gradient descent** is an optimization algorithm that estimates the error gradient for the current state of the model using examples from the training dataset, then updates the weights of the model using the [back-propagation of errors algorithm](https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/), referred to as simply backpropagation.

The amount that the weights are updated during training is referred to as the step size or the “*learning rate*.”

**The learning rate** is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0.

The learning rate controls how quickly the model is adapted to the problem. Smaller learning rates require more [training epochs](https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/) given the smaller changes made to the weights each update, whereas larger learning rates result in rapid changes and require fewer training epochs.

A learning rate that is too large can cause the model to converge too quickly to a suboptimal solution, whereas a learning rate that is too small can cause the process to get stuck.

The challenge of training deep learning neural networks involves carefully selecting the learning rate. It may be the most important hyperparameter for the model.

*The learning rate is perhaps the most important hyperparameter. If you have time to tune only one hyperparameter, tune the learning rate.*

**Q12. What is bias-variance trade off in machine learning?**

**Ans.**

The goal of any supervised machine learning algorithm is to achieve low bias and low variance. In turn the algorithm should achieve good prediction performance. You can see a general trend in the examples above:

* Linear machine learning algorithms often have a high bias but a low variance.
* Nonlinear machine learning algorithms often have a low bias but a high variance.

The parameterization of machine learning algorithms is often a battle to balance out bias and variance.

Below are two examples of configuring the bias-variance trade-off for specific algorithms:

* The k-nearest neighbors algorithm has low bias and high variance, but the trade-off can be changed by increasing the value of k which increases the number of neighbors that contribute t the prediction and in turn increases the bias of the model.
* The support vector machine algorithm has low bias and high variance, but the trade-off can be changed by increasing the C parameter that influences the number of violations of the margin allowed in the training data which increases the bias but decreases the variance.

There is no escaping the relationship between bias and variance in machine learning.

* Increasing the bias will decrease the variance.
* Increasing the variance will decrease the bias.

There is a trade-off at play between these two concerns and the algorithms you choose and the way you choose to configure them are finding different balances in this trade-off for your problem

In reality, we cannot calculate the real bias and variance error terms because we do not know the actual underlying target function. Nevertheless, as a framework, bias and variance provide the tools to understand the behavior of machine learning algorithms in the pursuit of predictive performance.

**Q13. What is the need of regularization in machine learning?**

**Ans.**

**Overfitting** is a phenomenon that occurs when a Machine Learning model is constraint to training set and not able to perform well on unseen data.

Regularisation is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.  
The commonly used regularisation techniques are :

1. L1 regularisation
2. L2 regularisation
3. Dropout regularisation

This article focus on L1 and L2 regularisation.

A regression model which uses **L1 Regularisation**technique is called **LASSO(Least Absolute Shrinkage and Selection Operator)** regression.  
  
A regression model that uses **L2 regularisation** technique is called **Ridge regression**.  
**Lasso Regression** adds *“absolute value of magnitude”* of coefficient as penalty term to the loss function(L).

https://media.geeksforgeeks.org/wp-content/uploads/20200705212608/L1regularisation1-300x23.png

**Ridge regression** adds “*squared magnitude*” of coefficient as penalty term to the loss function(L).

https://media.geeksforgeeks.org/wp-content/uploads/20200705212953/L2regularisation-e1593964838310-300x29.png

**Q14. Differentiate between Adaboost and Gradient Boosting**

**Ans.**

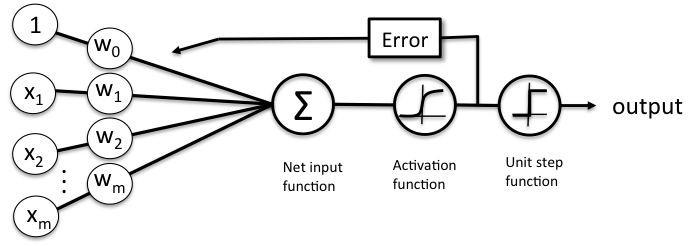
|  |  |
| --- | --- |
| **Gradient boosting** | **Adaptive Boosting** |
| This approach trains learners based upon minimising the loss function of a learner (i.e., training on the residuals of the model) | This method focuses on training upon misclassified observations. Alters the distribution of the 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 incase 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. |

**Q15. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?**

**Ans.**

The short answer is: Logistic regression is considered a generalized linear model because the outcome **always** depends on the **sum** of the inputs and parameters. Or in other words, the output cannot depend on the product (or quotient, etc.) of its parameters!

Let’s recapitulate the basics of logistic regression first, which hopefully makes things more clear. Logistic regression is an algorithm that learns a model for binary classification. A nice side-effect is that it gives us the *probability* that a sample belongs to class 1 (or vice versa: class 0). Our objective function is to minimize the so-called logistic function Φ (a certain kind of sigmoid function); it looks like this:



Logistic regression is known and used as a linear classifier. It is used to come up with a hyper*plane* 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*. Robust and efficient implementations are readily available (e.g. scikit-learn) to use logistic regression as a linear classifier.