MACHINE LEARNING – WORKSHEET 3

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

Ans- **Linear Kernel**is used when the data is linearly separable, that is, it can be separated using a single 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.

**RBF** **kernel** is a stationary kernel. It is also known as the “squared exponential” kernel. It is parameterized by a length scale parameter l>0, which can either be a scalar (isotropic variant of the kernel) or a vector with the same number of dimensions as the inputs X (anisotropic variant of the kernel).

**polynomial kernel** is a [kernel function](https://en.wikipedia.org/wiki/Kernel_function) commonly used with [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine) (SVMs) and other [kernelized](https://en.wikipedia.org/wiki/Kernel_trick) models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

1. 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-

1. 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-  **The explained sum of squares** (ESS), alternatively known as the model sum of squares or sum of squares due to, is a quantity used in describing how well a model, often a [regression model](https://en.wikipedia.org/wiki/Regression_analysis), represents the data being modeled.

It (ESS) is the sum of the squares of the deviations of the predicted values from the mean value of a response variable.

**The residual sum of squares** (RSS), also known as the sum of squared residuals (SSR) or the sum of squared estimate of errors (SSE), is the [sum](https://en.wikipedia.org/wiki/Summation) of the [squares](https://en.wikipedia.org/wiki/Square_(arithmetic)) of [residuals](https://en.wikipedia.org/wiki/Errors_and_residuals_in_statistics) .It is a measure of the discrepancy between the data and an estimation model. A small RSS indicates a tight fit of the model to the data. It is used as an [optimality criterion](https://en.wikipedia.org/wiki/Optimality_criterion) in parameter selection and [model selection](https://en.wikipedia.org/wiki/Model_selection).

**The total sum of squares** (TSS) is a quantity that appears as part of a standard way of presenting results of such analyses.

[Total sum of squares](https://en.wikipedia.org/wiki/Total_sum_of_squares) ( TSS ) = explained sum of squares (ESS)+ [residual sum of squares](https://en.wikipedia.org/wiki/Residual_sum_of_squares) (RSS).

1. What is Gini –impurity index?

Ans- Gini index measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.

A Gini Index of 0.5 denotes equally distributed elements into some classes.

Example of Gini Index:

|  |  |  |  |
| --- | --- | --- | --- |
| **Part Trend** | **Open Interest** | **Trading volume** | **Return** |
| Positive | Low | High | Up |
| Negative | High | Low | Down |
| Positive | Low | High | Up |
| Negative | High | High | Up |
| Positive | Low | High | Down |
| Negative | Low | Low | Down |
| Positive | High | High | Down |
| Negative | Low | High | Down |
| Positive | Low | Low | Down |
| Negative | High | High | Up |

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

Ans- Yes its is unregularized prone to overfitting. The Decision tree tends to overfit since at each node, it will make the decision among a subset of all t he features(columns), so when it reaches a final decision, it is a complicated and long decision chain. Only if a data point satisfies all the rules along this chain, the final decision can be made. This kind of specific rules on training dataset make it very specific for the training set, on the other hand, cannot generalize well for new data points that it has never seen. Especially when your dataset has many features(high dimension), it tends to overfit more.

There are parameters to control how deep your tree will grow, how many leaf node your tree will have when you grow a tree and so on. These parameters can give you the chance to prevent your tree from overfitting. However, in general, decision tree tends to grow deep to make a decision which makes it specific and complicated, as a result tends to overfit.

There are also concepts like 'unstable' and 'high variance' to describe this characteristic for tree models. 'High variance' and 'unstable' mean that the model is easy to change, say, your model could be completely different if the input changes a little bit.

If a model is too easy to change according to the input data, too flexible, 'high variance' , can fit every input dataset too well, makes it hard to generalize a bigger idea about the data trend. This also explains why overfit.

1. What is an ensemble technique in machine learning?

Ans- Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model.

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model.

Ensemble methods can be divided into two groups:

* sequential ensemble methods where the base learners are generated sequentially (e.g. AdaBoost).  
  The basic motivation of sequential methods is to exploit the dependence between the base learners. The overall performance can be boosted by weighing previously mislabeled examples with higher weight.
* parallel ensemble methods where the base learners are generated in parallel (e.g. Random Forest).The basic motivation of parallel methods is to exploit independence between the base learners since the error can be reduced dramatically by averaging.

1. What is the difference between Bagging and Boosting techniques?

Ans-

**Bagging :** Bagging is also known as bootstrap aggregating sits on top of the majority voting principle. The samples are bootstrapped each time when the model is trained. When the samples are chosen, they are used to train and validate the predictions. The samples are then replaced back into the training set. The samples are selected at random. This technique is known as bagging. To sum up, base classifiers such as decision trees are fitted on random subsets of the original training set. Subsequently, the individual predictions are aggregated (voting or averaging etc.). The final results are then used as predictions. It reduces the variance of a black box estimator. Due to this the chances of overfitting is ruled out.

**Boosting:** The concept of Adaptive Boost revolves around correcting previous classifier mistakes. Each classifier gets trained on the sample set and learns to predict. The misclassification errors are then fed into the next classifier in the chain and are used to correct the mistakes until the final model predicts accurate results. When a weak-classifier misclassifies a training sample, the algorithm then uses these very samples to improve the performance of the ensemble.

1. 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 (bagging) to sub-sample data samples used for training.

1. What is K-fold cross-validation?

Ans- 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.

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

Ans- In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), hyperparameter optimization or tuning is the problem of choosing a set of optimal [hyperparameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)" \o "Hyperparameter (machine learning)) for a learning algorithm. Parameters which define the model architecture are referred to as hyperparameters and thus this process of searching for the ideal model architecture is referred to as hyperparameter tuning.

It is done to choose a set of optimal [hyperparameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)" \o "Hyperparameter (machine learning)) for a learning algorithm, also can increase the model performance.

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

Ans- 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.

When the learning rate is too large, gradient descent can inadvertently increases rather than decreases the training error.

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

Ans- Bias is the simplifying assumptions made by the model to make the target function easier to approximate. Variance is the amount that the estimate of the target function will change given different training data. Trade-off is tension between the error introduced by the bias and the variance.

1. What is the need of regularization in machine learning?

Ans- Regularizations are techniques used to reduce the error by fitting a function appropriately on the given training set and avoid overfitting. Also this technique discourages learning a more complex or flexible model.

1. Differentiate between Adaboost and Gradient Boosting

Ans-

The main differences therefore are that Gradient Boosting is a generic algorithm to find approximate solutions to the additive modeling problem, while AdaBoost can be seen as a special case with a particular loss function. Hence, gradient boosting is much more flexible.

Second, AdaBoost can be interepted from a much more intuitive perspective and can be implemented without the reference to gradients by reweighting the training samples based on classifications from previous learners.

In Gradient Boosting, shortcomings are identified by gradients.

In Adaboost, shortcomings are identified by high-weight data points.

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

Ans- 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.