Support Vector Machine
– Questions & Answers

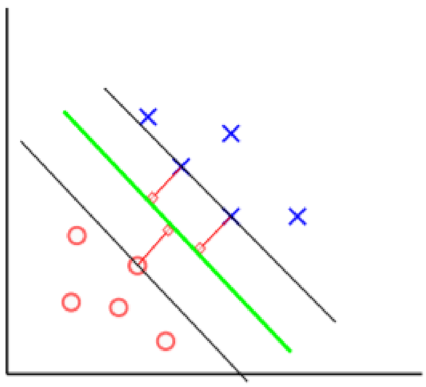
Assignment Subjective Questions

Author	**Rahul Kumar**
Date	**21-July-2019**
Course	**PGDDS**
Batch	**Dec 2018**

1. How is Soft Margin Classifier different from Maximum Margin Classifier?

→ A margin classifier is a classifier which is able to give an associated distance from the decision boundary for each example. For instance, if a linear classifier (e.g. perceptron or linear discriminant analysis) is used, the distance of an example from the separating hyperplane is the margin of that example.
The notion of margin is important in several machine learning classification algorithms, as it can be used to bound the generalization error of the classifier. These bounds are frequently shown using the VC dimension. Of particular prominence is the generalization error bound on boosting algorithms and support vector machines.

Maximizing the distance of the separator to the examples seems the right choice (actually supported by PAC learning theory). This means that only the nearest instances to the separator matter (the rest can be ignored)



The constraint of maximizing the margin of the line that separates the classes must be relaxed. This is often called the soft margin classifier. This change allows some points in the training data to violate the separating line.

Soft margin is extended version of hard margin SVM. Hard margin SVM can work only when data is completely linearly separable without any errors (noise or outliers). In case of errors either the margin is smaller or hard margin SVM fails. On the other hand soft margin SVM was proposed by Vapnik to solve this problem by introducing slack variables. As for as their usage is concerned since Soft margin is extended version of hard margin SVM so we use Soft margin SVM.

2. What does the slack variable Epsilon (ε) represent?

➔ Slack variables are defined to transform an inequality expression into an equality expression with an added slack variable. The slack variable is defined by setting a lower bound of zero (>0). Introducing a slack variable replaces an inequality constraint with an equality constraint and a non-negativity constraint on the slack variable.

Slack variables are used in particular in linear programming. As with the other variables in the augmented constraints, the slack variable cannot take on negative values, as the simplex algorithm requires them to be positive or zero.

- If a slack variable associated with a constraint is zero at a particular candidate solution, the constraint is binding there, as the constraint restricts the possible changes from that point.
- If a slack variable is positive at a particular candidate solution, the constraint is non-binding there, as the constraint does not restrict the possible changes from that point.
- If a slack variable is negative at some point, the point is infeasible (not allowed), as it does not satisfy the constraint.

3. How do you measure the cost function in SVM? What does the value of C signify??

➔ In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the **cost function**. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

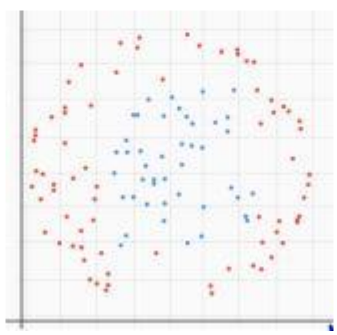
$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights. When there is no misclassification, i.e our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

When there is a misclassification, i.e our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.

The **C parameter** tells the SVM optimization how much we want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, we should get misclassified examples, often even if our training data is linearly separable.

4. How Given the above dataset where red and blue points represent the two classes, how will you use SVM to classify the data?



➔ The above dataset where red and blue points represent the two classes can be classify using SVM with RBF kernel. This could be further tune with hyperparameter to optimize separation. Also we can optimized accuracy and precision.

5. What do you mean by feature transformation?

➔ Feature transformation (FT) refers to create new features using the existing features. These new features may not have the same interpretation as the original features, but they may have more discriminatory power in a different space than the original space. This can also be used for feature reduction. FT may happen in many ways, by simple/linear combinations of original features or using non-linear functions. Some common techniques for Feature Transformation are:

- Scaling or normalizing features within a range, say between 0 to 1 or -1 to 1
- Principle Component Analysis and its variants

- Random Projection
- Neural Networks
- SVM also transforms features internally
- Transforming categorical features to numerical

Feature transformation is also known as feature engineering, which may help in improving model performance.

-------------------------------- End of File ---------------------------------