# CHAPTER 5

## Conclusion and Future Works

This thesis proposed a family of supervised learning machine which utilizes hyperellipsoids as a set of predefined functions to create a decision boundary. Specifically, we proposed two novel supervised classifiers, neSVDD and TESVM. neSVDD is named after "ellipsoidal support vector data description with negative examples". It is particularly designed for solving one-class classification problems, while TESVM or "twin hyperellipsoidal support vector machine" is designed for two-class classification problems. Both neSVDD and TESVM are, in fact, the attempts on extending SVDD and THSVM from the spherical descriptive domains to the ellipsoidal ones with the introduction of the minimum volume covering ellipsoid (MVCE). neSVDD aims at using the MVCE formulated with soft margins as a data description tool by trying to cover the target data but excluding outliers at the same time. Particularly, the inclusion of negative examples into the soft-margin MVCE is one main contribution of this thesis. Although TESVM is rather close to neSVDD, its formulation and objective is different. TESVM tries to construct one hyperellipsoid around one class while also trying to make it as far as possible from the other class. In fact, TESVM is the hyperellipsoid version of THSVM and TWSVM.

In addition, as inspired by other kernel learning machines, like SVM, the other add-on important ingredient of the proposed methods is the uses of kernel methods. This is very indispensable ability for a learning machine to create more sophisticated decision boundary to cope with real-world data. From the fact that MVCE cannot directly apply kernel tricks by simply replacing any inner product pair with a kernel function like in SVM, TWSVM, and THSVM because the formulation of MVCE consists of the outer product. Therefore, this thesis also proposed the so-call empirical feature mapping to help construct hyperellipsoids in the kernel feature space.

The experimental results presented in Chapter 4 show that ellipsoids provide more refined boundary than spheres as can be seen from the improved results when comparing SVDD with eSVDD, nSVDD with neSVDD, and THSVM with TESVM. In addition, the introduction of negative examples helps enhance the overall results in the case of neSVDD vs. eSVDD. In the case of TESVM, TESVM with linear kernel significantly provides an improvement over THSVM with linear kernel. Although such an improvement may not be so large with the use of RBF kernel, it can still be considered an improvement. TESVM with both either linear or RBF kernel also offers better average accuracy than SVM and TWSVM.

In addition, to further validate the performance of TESVM on larger datasets, an instance reduction scheme proposed by A. Yodjaiphet [85] is also introduced as a data preprocessing step to reduce the number of training examples prior to training a TESVM model. The experimental results also show that TESVM is superior to both THSVM and TWSVM which are from the same family of twin support vector classifiers. Although the overall performance of TESVM is weaker than of SVM, we observe that TESVM provides the best classification accuracy in many datasets. In fact, the results from approximately 4 out of 10 datasets are better than from SVM.

For the future work, the use of empirical feature map indeed opens a possibility to perform kernel optimizations to find the most suitable kernel for each dataset [75]. In addition, due to the complexity of solving a semidefinite program, an important future work is to improve the computational speed of neSVDD and TESVM. The instance reduction technique suggested in this thesis is not the only viable option. For example, it is possible to apply the dual reduced Newton algorithm [41] which also utilizes the structure of MVCE to solve the optimization. Another possibility is to apply active-set strategies [41] or a scheme like that in [63] to select only some subsets of data which are important for the creation of hyperellipsoids. Moreover, it is challenging to determine the best method for hyperparameter tuning and also to understand the effects of each hyperparameter.