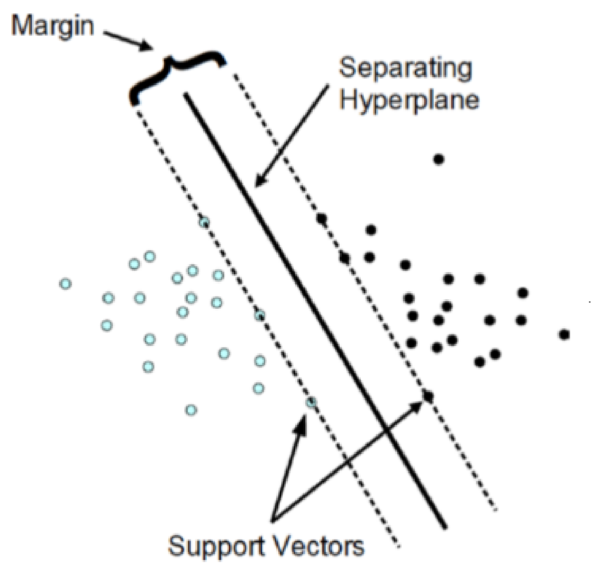
Support Vector Machines Implementation Issues

Support Vector Machines (SVMs) are supervised learning models with learning algorithms that analyze data which can be used in both classification and regression problems, mostly classification problems. For a training dataset, each point belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other. They are based on finding a hyperplane that best divides the datasets into two classes. SVMs are widely used in text and hypertext classification, image classification, recognition of hand-written characters, and in biology and science for classification.

For instance, Sam wants to implement SVMs in text classification. First, Sam transforms a piece of text into a vector of numbers, so he can run SVM with them. We can use the Bag of Words model to do so, and for each word in the bag he will have a feature. This method boils down to calculating the frequency of each word and dividing that number by the total number of words. Sam then provides this to the SVM as the training dataset. Next, Sam must choose a kernel function for our model. The linear kernel is used more commonly since the presence of many features results in the nonlinear kernels overfitting the data. Lastly, Sam takes a set of labeled texts, converts them to vectors using frequencies, and feeds them to the algorithm, which uses Sam's chosen kernel function. If Sam's model is provided with new unlabeled text he wants to classify, he will convert it into a vector and give it to the model, which will output the tag of text.



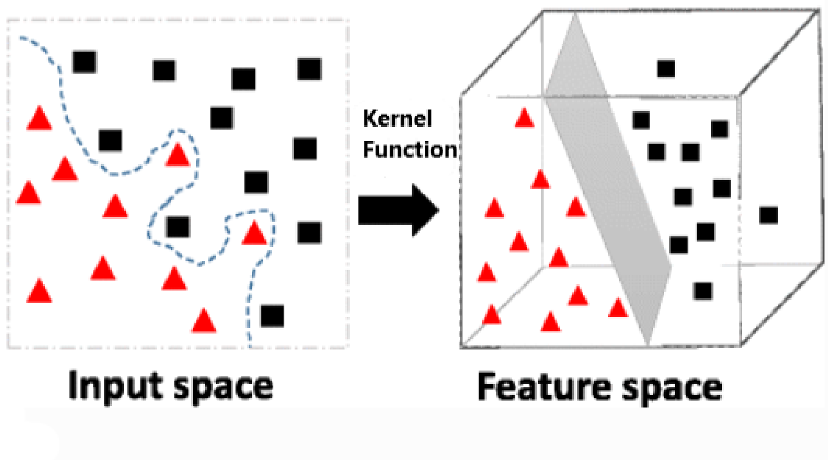
The above figure depicts how a hyperplane separates the points into two classes. The points lying on the margin are called support vectors. The hyperplane is chosen by maximizing the margin between the closest points of the two classes

The following are the implementation issues associated with SVMs:

- The biggest disadvantages are choosing appropriately hyper parameters of the SVM that will allow for sufficient generalization performance. There are several key parameters that need to be set correctly to achieve the best classification results for any given problem. Users will have to experiment with a number of different parameter settings to receive satisfactory results.
- One of the biggest limitation associated with SVMs lies in the choice of the kernel. Determining the best kernel type for a specific dataset may require a tedious trial-and-error approach, especially when the dataset is rather large with a lot of variables (Efraim).
- In Natural Language Processing, structured representations of

text yield better performances. However, SVMs are unable to accommodate such structures (word-embeddings) since they use the Bag-of-words representation which loses sequential information and leads to worse performances.

- SVMs are less effective in the presence of noisy data and overlapping points in data. This results in a problem in drawing a clear hyperplane without misclassification.
- The most serious problem with SVMs is the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks. The minimum time complexity for training SVMs is O(n2). Hence, SVMs are not suitable to implement for large datasets. Datasets which have a clear classification boundary will function best with SVMs and will have limitations otherwise.
- SVMs use a-priori chosen kernel functions to compute similarities between input vectors. An issue with this is that using the best kernel function is important, but kernel functions are not very flexible (Wiering, Schutten, Millea, Meijster, Schomaker).
- The standard SVM only has one adjustable layer of model parameters. It is preferred to use deep architectures instead of such 'shallow models' (Wiering, Schutten, Millea, Meijster, Schomaker).



The figure above depicts the transformation of non-linear data to a linear one in higher dimensional feature space, using the kernel function

- The SVM tuning process consists of a gamma parameter which influences the points near or far away from the hyperplane. If the gamma value is low, the model is too constrained and will include all the points in the training dataset without really capturing the shape.
- SVMs can be abysmally slow in test phase even though they have good generalization performance. This makes them less useful in situations that require a fast response (Efraim).
- SVM is a binary classifier. To do a multi-class classification, pair-wise classifications need to be used (one class against all other, for all classes).
- SVMs have high algorithmic complexity and extensive memory requirements due to the use of quadratic programming.
- SVMs move the problem of over-fitting from optimizing the parameters to model selection. Sadly, kernel models can be very sensitive to over-fitting the model selection criterion.

- The presence of discrete data provides a problem. It is not obvious how to perform efficient optimization on discrete kernel weights (Lewis, Jebara, Noble, 2006).

Efraim, T. Decision Support and Business Intelligence Systems. Retrieved from: https://books.google.co.in

Lewis, D. Jebara, T. Noble, W. Support Vector Machine Learning from Heterogeneous Data: An Empirical Analysis Using Protein Sequence and Structure. 2006. Retrieved from: https://academic.oup.com

Wiering, M. Schutten, M. Millea, A. Meijster, A. Schomaker, L. Deep Support Vector Machines for Regression problems. Retrieved from: http://www.ai.rug.nl/~mwiering