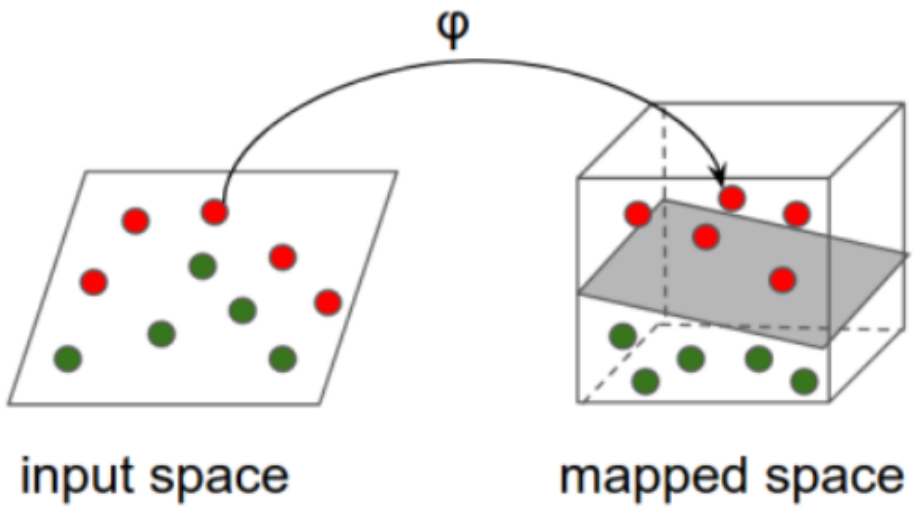
Kernel Implementation Issues

Kernels methods are a class of algorithms for pattern analysis, whose best-known member is the Support Vector Machine (SVM). A kernel-based algorithm is a non-linear version of a linear algorithm where the data has been previously (non-linearly) transformed to a higher dimensional space in which we only need to compute the inner products using the kernel function. Kernel methods are used in geo-statistics, 3D reconstruction, bioinformatics, information extraction, and handwriting recognition. Pattern recognition problems in Bioinformatics are characterized by high dimensional and noisy data, by the need to insert prior knowledge in the system and to merge heterogeneous sources of data. The class of Kernel Methods has been designed to work in such conditions and delivers state of the art performance in many bioinformatics applications. SVMs, have been used to analyze DNA microarray data, sequence data, phylogenetic information, promoter region information, and so on. One of the main features of these systems is their modularity: they are formed by a general-purpose learning module and by a problem specific kernel. For example, a classification module can be coupled with a sequence matching kernel to obtain a protein sequences classifier, as demonstrated using SVMs for protein homology detection.

The following are the implementation issues associated with Kernels:

- Kernel methods typically construct a kernel matrix $K \in R^{N \times N}$ with N as the number of training instances. The complexity of the kernel methods is therefore a function of the number of training instances, and hence for problems with very large N the complexity is currently prohibitive. For example, for an SVM the

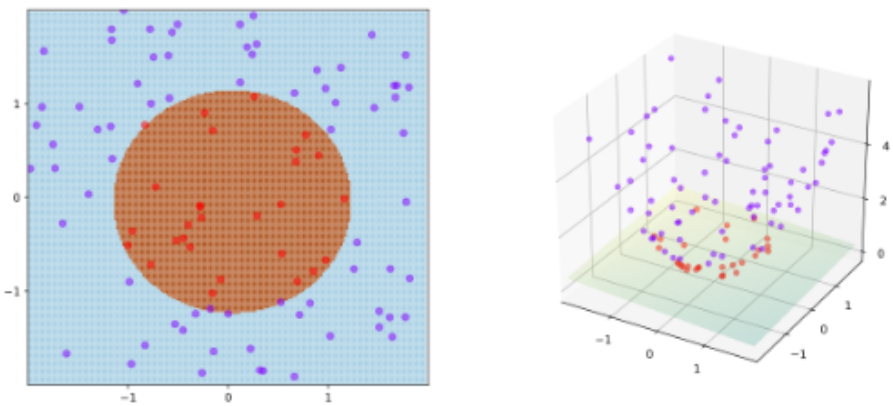
training complexity is between O(N^2) and O(N^3).



The figure above depicts how a non-linear map transforms the input space to another feature space where the features are linearly separable and then apply a linear model on the mapped features

- Kernel methods also have a serious statistical limitation. For any finite m, the prediction accuracy depends critically on the chosen distance function, especially for regression. When there are more than a few predictor variables, even the largest datasets produce a very sparse sampling in the corresponding n-dimensional predictor variable space. This is due to the 'curse of dimensionality'. For the kernel methods to perform well, the distance function must be carefully matched to the unknown target function, and the procedure is not very robust to mismatches (Soman, Logananthan, Ajay, 2009).
- Kernels methods are known to be incomprehensible. It is difficult to understand what the kernel method has learned, aside from a list of elements that make up the Support Vector.
- Since there is no model, kernel methods provide no easily

understood model summary. Hence, they cannot be easily interpreted. There is no way to discern how the function Fm(x) depends on the respective predictor variables x (Soman, Logananthan, Ajay, 2009).



The figure above depicts SVM with kernel given by $\phi((a,b)) = (a, b, a^2 + b^2)$ and thus K(x, y) = x . y + x^2 y^2. The training points are mapped to a 3-dimensional space where a separating hyperplane can easily be found

- Kernel methods find it hard to generalize multi-class problems. To do a multi-class classication, pair-wise classications need to be used (one class against all other, for all classes).
- Kernel method does not perform as well as deep learning. The accuracy of the systems constructed using kernel methods are now outperformed on many domains by the deep learning approach.
- Kernel methods face a pre-image problem. They lack an easy way to map inversely from the feature space back to the input data space. This issue comes up in kernel Principal Component Analysis (kernel PCA), where one wants to find the pattern in the input space corresponding to a principal component in the feature space (Hsieh, 2009).

- Kernel methods produce a 'black-box' prediction machine. To make each prediction, the kernel method needs to examine the entire database. This requires enough random-access memory to store the entire data set, and the computation required to make each prediction is proportional to the training sample size m. For large datasets, this is very slow (Soman, Logananthan, Ajay, 2009).

Hsieh, W. Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels. 2009. Retrieved from: https://books.google.co.in

Soman, K. Loganathan, R. Ajay, V. Machine Learning with SVM and Other Kernel Methods. 2009. Retrieved from: https://books.google.co.in