SOME PRACTICAL CONSIDERATIONS ON USING SVMS AND OTHER KERNEL-BASED METHODS

Sékou Tidiani COULIBALY
Advisor: Gábor HORVÁTH

I. Introduction

Statistical learning theory opened the era of kernel methods, and many algorithms related to classical pattern recognition are being rewritten (extended) to their kernel formulations. This paper puts emphasis on practical considerations that may have influence on the performance goal of SVMs or any other kernel based procedure, i.e. maximization of generalization ability and capacity control.

Unlike multilayer neural network support vector machines use direct functions to implement its goals. In two-class, $x_i \in \Re^n$ instances, classification problems and in the linearly separability case with M training samples $(x_{i,}y_i) \in X \times \{\pm 1\}$, we use $W^T x_i + b \geq 1$ for $y_i = 1$ and $W^T x_i + b \leq -1$ for $y_i = -1$ decision functions. The hard margin SVM implementing the optimal separating hyperplane, having W as orthogonal vector, can be obtained by solving the quadratic problem given by Eq.(1).

$$Q(W,b,\alpha) = \tfrac{1}{2} W^T W - \sum_{i=1}^{M} \alpha_i \{ y_i (W^T x_i + b) - 1 \} . \tag{1}$$

In Eq.(1), α_i associated with each training sample are the nonnegative Lagrange multipliers.

Many problems of practical interest fall in the case of nonlinearly separability, then Eq.(1) has no feasible solution and the hard-margin SVM is unsolvable. The separating hyperplane is then constructed following the principles of structural risk minimization (SRM) [1]. The algorithm leads to the quadratic optimization problem as shown below:

$$Q(W,b,\alpha,\xi) = \tfrac{1}{2} W^T W + \tfrac{1}{2} C \sum_{i=1}^{M} \xi_i^p - \sum_{i=1}^{M} \alpha_i \{ y_i (W^T g(x_i) + b) - 1 + \xi_i \} . \tag{2}$$

The optimal solution must satisfy either α_i=0 or condition of Eq.(3).

$$y_i \{ \sum_{j=1}^{M} \alpha_j y_j (K(x_j, x_i) + \delta_{ij} . \frac{1}{C}) + b \} - 1 = 0 . \tag{3}$$

Here, δ_{ij} stands for Kronecker's delta function; it is 1 for $i = j$, otherwise 0. In Eq.(2), $g(x)$ is typically a nonlinear mapping function to the feature space; ξ the slack variables gives chance of having feasible solution to the optimization problem. When p value takes 1 or 2 we have L1 SVM or L2 SVM, with the former having the associated computations stable and an enhanced generalization [6]. From the above equation it can be seen that to construct an efficient support vector machine one can use a positive definite function $K(x_j, x_i)$ called kernel function. Different types of kernel are applied to solve kernel specific problems. All the above equations show that in SVM (kernel methods) paradigm there are two major stages. The first is the regularization frame to formulate the optimization problem to solve and define the parameter(s) controlling the margin size during training the SVM. The second is the SVM model selection and the choice of the kernel function.

II. Regularization

Minimizing the "pure" empirical risk can lead to numerical instabilities and bad generalization performance. Regularisation is a possible way to avoid these problems and a procedure to restrict the class of admissible solutions, for instance to a compact set. This technique was introduced by

Tikhonov and Arsenin [1, 2] for solving inverse problems and has since been applied to learning problems hence they are generally ill-posed.

In various SVM algorithms the parameter *C* controls the trade-off between complexity and portion of data samples that are nonseparable; it determines implicitly the size of margin. This margin embeds the criterion of falsifiability [1, 4], formulated by K. Popper as necessary condition for a true theory to be falsifiable by certain observations, facts or data samples. The practice (principle) using the above idea, the SRM can be formulated as follow:

- Minimize the total empirical risk for data samples lying inside the margin.
- Achieve maximum separation (margin) between training data samples that are correctly classified.

The coefficient of the penalty term, *C*, also needs to be defined by the user. Again, there is no easy method for selecting its value aside from evaluating the resulting model's performance on a validation set.

III. Kernel function selection

The choice of kernel function is crucial in all kernel-based algorithms. The kernel function is seemed to be a prior knowledge that is available about a task in addition to the empirical observations. Although the question of how to choose the "best" kernel function for a given dataset is often posed, it has no "good" answer on top of this and it may be crucial for success.

A more formal metric for choosing the best kernel is provided by the upper bound on the VC dimension [1, 2, 3]. However, this remain an assumption even though the VC dimension describes the complexity and flexibility of the kernel, it does not provide practical proof that the chosen kernel is the "best" one. The choice can be validated by methods such as cross-validation.

Originally SVMs are based on fixed-length input data. For non-vector based applications there are two approaches to handle in SVMs or other kernel algorithms. The first is to extract vector-based features. The second is to create feature-specific kernel functions.

The list of existing kernels in literatures on the field is growing: linear, polynomial, radial basis, string, three-layer neural network, Hausdorff, and histogram intersection kernel functions [3, 4].

Beside the choice of the proper kernel function, may arise the task of kernel preconditioning. In fact, when the number of input variables is very large numerical problems make difficult the training of SVMs. Some examples of kernel preconditioning (normalizing) are given in [5], the handling of the bias term in polynomial and RBF kernels are also discussed.

IV. Conclusion

A great number of solutions to problems that is related to classical pattern recognition have adopted kernel paradigm: Kernel PCA and Kernel Feature Analysis, Regularized Principal Manifolds, estimation of the support of a distribution, Kernel Discriminant Analysis and Relevance Vector Machines [1, 3], are involved in the kernel selection problem and other here non mentioned algorithms need the development application specific kernel functions.

The knowledge about various regularization (function penalisation) techniques can be useful in solving kernel-based problems [3].

References

[1] V. N. Vapnik, "Estimation of Dependencies Based on Empirical Data", Springer, 2006

[2] R. Herbrich, "Learning Kernel Classifiers: theory and algorithms", The MIT Press Cambridge, 2002.

[3] B. Schölkopf, A. J. Smola, "Learning with Kernels: Support vector Machines, Regularization, Optimization, and Beyond", The MIT Press, 2002.

[4] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

[5] S. Abe, "Support Vector Machines for Pattern Classification", Springer-Verlag, New York, 2005.