

A Short Note About the Application of Polynomial Kernels with Fractional Degree in Support Vector Learning

Rolf Rossius¹, Gerard Zenker², and Andreas Ittner¹

¹ Department of Computer Science
Artificial Intelligence Research Group
Chemnitz University of Technology
D-09107 Chemnitz

{ros,ait}@informatik.tu-chemnitz.de

² Department of Mathematics
Mathematical Optimization Research Group
Chemnitz University of Technology
D-09107 Chemnitz
zenker@mathematik.tu-chemnitz.de

Abstract. In the middle of the 90's a fundamental new Machine Learning approach has been developed by V. N. Vapnik: The Support Vector Machine (SVM). This new method can be regarded as a very promising approach and is gradually getting more attention in the fields where neural networks and decision tree methods are applied. Whilst neural networks may be (correctly or not) considered to be well understood and in wide use, Support Vector Learning has some rough edges in theoretical details and its inherent numerical tasks prevent an easy application in practice.

The paper picks up one important aspect - the use of fractional degrees on polynomial kernels in the SVM - discovered in the course of a practical realization of the algorithm. Fractional degrees on polynomial kernels broaden the capabilities of the SVM in a fundamental way and offer the possibility to deal with feature spaces of infinite dimension. We introduce a trick to simplify the quadratic programming problem, as the core of the SVM.

1 Introduction

Typical and well known representatives of classification and prediction in the field of Machine Learning are neural networks and methods for generation different kinds of decision trees. An innovative, but still relatively unknown learning approach is the Support Vector Machine (SVM) developed by V. N. Vapnik in the middle of the 90's. Support Vector Learning [IRZ98] here is not only another approach of learning techniques but can also be regarded as a fundamental new philosophy in the area of Machine Learning.

The underlying principle of the SVM is the principle of the *Structural Risk Minimization* (SRM) [Vap95]. In contrast to a pure minimization of the empirical

risk the SRM is based on the “idea of the simplicity” and unifies *Empirical Risk Minimization* and the problem of *Model Selection*. The searched binary classifier for the problem

$$(x_1, y_1), \dots, (x_l, y_l), x_i \in \mathbf{R}^n, y_i \in \{+1, -1\}, \quad (1)$$

has to be a function from the set

$$\{f_\alpha : \alpha \in \Gamma\}, f_\alpha : \mathbf{R}^n \rightarrow \{+1, -1\}, x \mapsto y, \quad (2)$$

of functions and should reflect the real inherent essence of the given learning problem. This essence could be regarded as the simplest (in some sense) separation of the feature space. The simplicity here will be formalized by means of the *VC dimension*, i. e. a measure about the considered set of feasible functions, e. g. the family of separating hyperplanes. The principle of SRM is enforced by controlled bounding of the VC dimensions of the set $\{f_\alpha : \alpha \in \Gamma\}$ and ensures the excellent ability of generalizing of the SVM. The underlying theory of the SRM should not be explained in detail in the presented paper. Instead we refer to [Vap95] which covers the SRM and the application in the SVM.

The separating hyperplane is characterized by ω and b :

$$\langle \omega, x \rangle + b = 0. \quad (3)$$

The distance between the hyperplane ω and the examples should be maximized, i. e. one has to solve a problem of mathematical programming. For the non-separable case slack variables $\xi_i \geq 0$ are introduced, which leads to:

$$\begin{cases} \frac{1}{2} \langle \omega, \omega \rangle + C \sum_{i=1}^l \xi_i \rightarrow \min \\ y_i [\langle \omega, x_i \rangle + b] \geq 1 - \xi_i & \forall i = 1, \dots, l \\ \xi_i \geq 0 & \forall i = 1, \dots, l, \end{cases} \quad (4)$$

where the parameter $C > 0$ controls the interrelationship between the accuracy of the classifier on the learning set and its ability of generalization, i. e. the accuracy on an unseen test set.

The vector ω , as the solution of (4), determines the optimal hyperplane. It can be expressed as a linear combination of only few examples of the whole learning data:

$$\omega = \sum_{i=1}^l \alpha_i y_i x_i = \sum_{SV} \alpha_i y_i x_i. \quad (5)$$

Support Vectors are such vectors x_i , which satisfy $y_i [\langle \omega, x_i \rangle + b] = 1$, i. e. which have a nonzero α_i and effectively contribute to the description of the separating hyperplane. Hence there exists the reducibility in (5) to a linear combination of support vectors. Less formally these support vectors could be viewed as the examples on the frontline guarding the own class against the examples of the other one and are essential for the concept has to be learnt.

Considering (5) one has to solve the following optimization problem:

$$\begin{cases} A^T \mathbf{1} - \frac{1}{2} A^T A A \rightarrow \max \\ \mathbf{0} \leq A \leq C \mathbf{1} \\ A^T Y = 0 \end{cases} \quad (6)$$

with $A = (\alpha_1, \dots, \alpha_l)$, $\mathbf{1} = (1, \dots, 1)$, and $Y = (y_1, \dots, y_l)$. The HESSE matrix A is constituted by the entries $A_{ij} = y_i y_j \langle x_i, x_j \rangle$ for $i, j = 1, \dots, l$ [CV95].

In the general case the linear separation in the original feature space will not provide a sufficient classifier. Here the mapping $\Phi : \mathbf{R}^n \rightarrow \mathbf{R}^N$ ($n \ll N$) comes in and the original feature space expands to a very high dimensional image space like:

$$\Phi : \mathbf{R}^n \rightarrow \mathbf{R}^N, \quad \Phi(x) = (1, \gamma_1 x_1, \dots, \gamma_n x_n, \gamma_{n+1} x_1^2, \dots, \gamma_k x_n^d). \quad (7)$$

Now the linear separation can take place in this space. An inverse transformation back into \mathbf{R}^n results in a non-linear separation in the original space of the task supplied features:

$$f(x) = \langle \omega, \Phi(x) \rangle + b. \quad (8)$$

It is not necessary to expand the feature space explicitly. One means to do the mapping implicitly is the usage of kernels $K(u, v)$ (respectively inner products). In this context the fundamental interrelation is:

$$K(u, v) = \langle \Phi(u), \Phi(v) \rangle. \quad (9)$$

The symmetric function $K(u, v)$ may be an inner product for the high dimensional image space, if the eigenvalues are positive. A sufficient condition is given in MERCER's theorem [Vap95]:

$$\int \int K(u, v) g(u) g(v) du dv > 0 \quad \forall g : \int g^2(u) du < \infty. \quad (10)$$

One of the rather simple types of such kernels are representable as

$$K(u, v) = (\langle u, v \rangle + 1)^d, \quad d = 1, 2, \dots \quad (11)$$

with degree d as an integer. Other choices may be

$$\begin{aligned} K(u, v) &= (\langle u, v \rangle)^d \\ K(u, v) &= e^{-\frac{\|u-v\|}{\sigma}}. \end{aligned}$$

A special kind of the kernel (11) should be examined in the presented work.

2 Polynomial Kernels with Fractional Degree

However a fixed chosen kernel $K(u, v)$ induces not only exactly one transformation but a manifold of possible Φ 's. Even the dimensionality of the image space \mathbf{R}^N is not determined. For instance (11), in the case of $d = 2$ and $n = 2$ one could have an explicit Φ :

$$\Phi(u) = (1, \sqrt{2}u_1, \sqrt{2}u_2, u_1^2, \sqrt{2}u_1u_2, u_2^2) \quad (12)$$

as well as

$$\Phi(u) = (1, u_1, u_1, u_2, u_2, u_1^2, u_1u_2, u_2u_1, u_2^2). \quad (13)$$

A question arises: Choose a kernel $K(u, v)$ satisfying MERCER's theorem – which is the space of smallest dimension for an image of Φ ? The answer for $d \in \mathbf{N}$ is $\binom{n+d}{d}$ (or equivalent $\binom{n+d}{n}$). While selecting an appropriate kernel there are huge jumps in the dimensionality of the image space. The capacity may be controlled by bounding the norm of the separating hyperplane, but another tuning parameter will still be there: The dimensionality.

Using a fractional exponent in the kernel (11) we encounter some interesting property: the inner product $\langle u, v \rangle$ may be less than -1 and we have a negative base to raise. Hence the HESSE Matrix A will not be real valued and therefore symmetric ($A^T = A$) anymore, but in fact contains complex entries. Nevertheless, A has the property of hermiticity ($A^* = A$).

This opens the possibility of a new formulation of (6). Because

$$A^T A A = A^T A^T A = A^T \frac{1}{2} (A + A^T) A = A^T \frac{1}{2} (A + (A^*)^T) A = A^T \text{Re}(A) A \quad (14)$$

we equivalently solve

$$\begin{cases} A^T \mathbf{1} - \frac{1}{2} A^T \text{Re}(A) A \rightarrow \max \\ \mathbf{0} \leq A \leq C \mathbf{1} \\ A^T Y = 0 \end{cases} \quad (15)$$

and got rid of the complex entries.¹

Exposing the kernel for arbitrary exponents d we get according to TAYLOR:

$$\begin{aligned} (\langle u, v \rangle + 1)^d &= 1 + d\langle u, v \rangle + \frac{d(d-1)}{2!} \langle u, v \rangle^2 + \frac{d(d-1)(d-2)}{3!} \langle u, v \rangle^3 \\ &+ \frac{d(d-1)(d-2)(d-3)}{4!} \langle u, v \rangle^4 + \dots \end{aligned} \quad (16)$$

Non-integer exponents do not terminate the serie like the integer ones, but the influence of high-order terms decreases nevertheless. Contrary to kernels with

¹ $\text{Re}(A)$ denotes the real part of its argument.

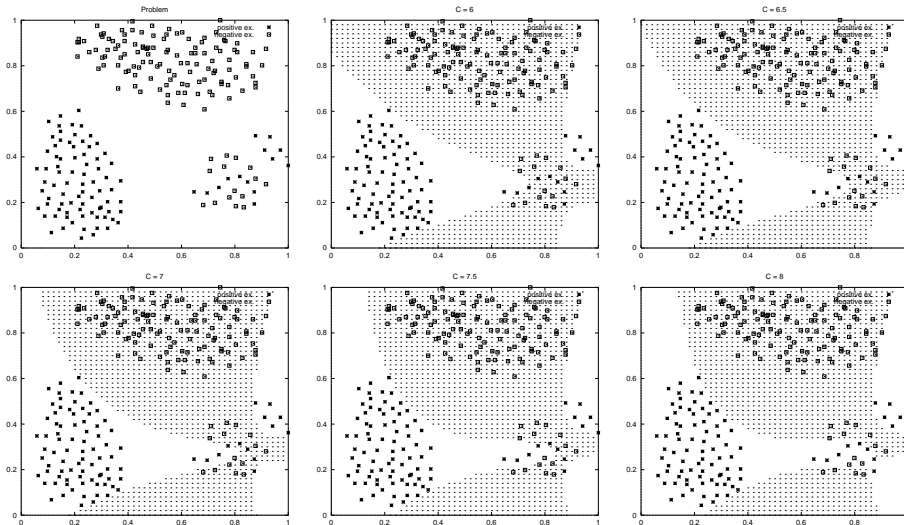


Fig. 1. Continuous variation of C

an integer exponent there are no mappings Φ corresponding to such a fractional exponent kernel which have an image space of finite dimension.

Fractional degrees allow a more continuous range of concepts. The resulting separating hyperplanes smoothly change the shapes with the exponent. This will be of importance especially for domains dealing with feature spaces which already cover tens, hundreds or more dimensions (e.g. recognition of graphical images), where a lower degree of a polynomial kernel is preferred. A simple problem [Fri93] in a two dimensional feature space is presented in Figure 1.

3 The “1/2 Trick”

Realizing the SVM as a whole, the solution of the quadratic optimization problem (quadratic programming, QP) – actually a serie of such, with different parameters – constitutes the real amount of work. Generally the QP task is for the most part determined by the calculation of function values, gradients (or its estimations). It is making more difficulties here because of the (potential) large HESSE matrix and its nonsparsity.

Choosing a kernel of the type $(\langle u, v \rangle + 1)^d$ with $d = m + \frac{1}{2}$ and $m \in \mathbf{N}$ the corresponding entry in the resulting HESSE matrix ($\text{Re}(A)$ in (15)) will be vanished for negative $(\langle u, v \rangle + 1)$.

The SVM algorithm selects a separating hyperplane according a criterion of sufficient values on the training examples as well as the minimization of the norm of the hyperplane. Unfortunately, the resulting shape of the function and thus the border between the predicted areas of both classes varies with uniform translations of the examples in the feature space. For instance the resulting

separation lines for different centered sets of the well known XOR problem ² is depicted in Figure 2. A second degree kernel is used.

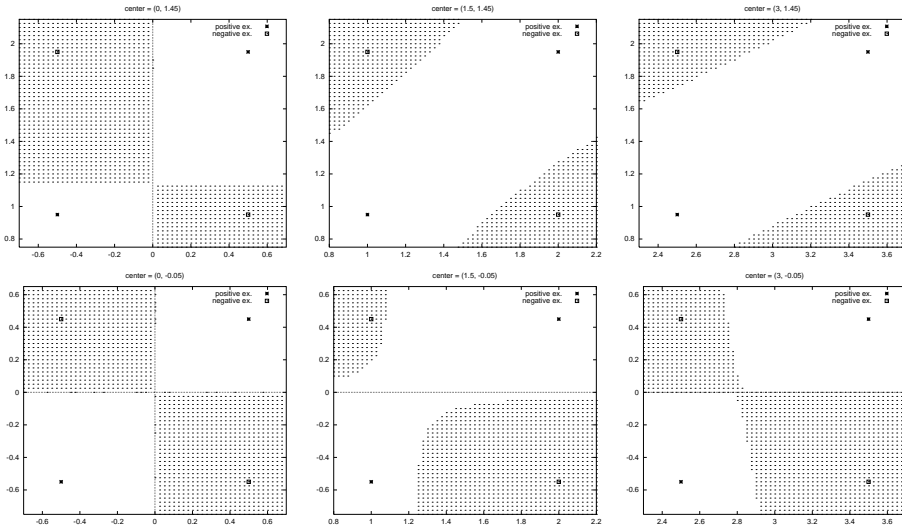


Fig. 2. Noninvariance of separation with respect to translation

Despite of the non-invariance against the uniform translation of the examples in the feature space, one could center the set into the origin of the co-ordinate system to obtain a sufficient obtuse angle between a large number of pairs of examples. This will result in a sparser HESSE matrix for the QP task. Up to fifty percent of the entries may be zeroed by means of this smart approach.

4 Summary

The Support Vector algorithm shows some promising properties but needs some refinement especially on the level of practical realization to soften the enormous effort to find the “simplest” explanation for a learning problem. Polynomial kernels with fractional degrees provides a broader range of concepts as well as a way to reduce the numerical effort to spend in the QP.

² $(x - \frac{1}{2}, y - \frac{1}{2}), (x + \frac{1}{2}, y + \frac{1}{2})$ are members of one class, while the two other examples $(x - \frac{1}{2}, y + \frac{1}{2}), (x + \frac{1}{2}, y - \frac{1}{2})$ belonging to a second class. The four points of the feature space are centered on (x, y) .

References

- [CV95] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- [Fri93] B. Fritzke. Growing cell structures – a self-organizing network for unsupervised and supervised learning. Technical Report 93-026, International Computer Science Institute, Berkeley, California, 1993.
- [IRZ98] A. Ittner, R. Rossius, and G. Zenker. Support Vector Learning. *submitted: KI-Themenheft Data Mining*, 1998.
- [Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.