A note on the universal approximation capability of support vector machines

Barbara Hammer and Kai Gersmann
LNM, Department of Mathematics/Computer Science, Universität Osnabrück, Germany, e-mail: {hammer, kai}@informatik.uni-osnabrueck.de

September 26, 2001

Abstract. The approximation capability of support vector machines (SVMs) is investigated. We show the universal approximation capability of SVMs with various kernels, including Gaussian, several dot product, or polynomial kernels, based on the universal approximation capability of their standard feedforward neural network counterparts. Moreover, it is shown that an SVM with polynomial kernel of degree $p-1$ which is trained on a training set of size p can approximate the p training points up to any accuracy.

Keywords: approximation, classification, kernel, support vector machine, universal approximation capability

1. Introduction

The support vector machine (SVM) proposed by Vapnik [4] is a particularly successful machine learning tool which learns an unknown regularity based on a finite set of examples. The area of applications ranges from time series prediction to text processing and various excellent tutorials and textbooks describe the method in detail [3, 11, 13, 19]. Depending on the actual kernel which is used for training, the SVM yields functions similar to functions implemented by standard feedforward neural networks (FNNs). However, the main idea behind the SVM is different: Roughly speaking, it implements a simple linear mapping or linear classifier together with a prior fixed nonlinear mapping in order to make data separable. This has two advantages: Training does not suffer from the problem of local minima; training can focus on optimality of the function with respect to its generalization ability. Both aspects often lead to better results compared to feedforward neural networks.

One key property of FNNs is their universal approximation capability: FNNs can approximate any reasonable mapping up to any desired accuracy if a sufficient number of neurons is available in the network. This property has first been investigated by Hecht-Nielsen [8]. Several approaches prove the universal approximation capability of FNNs with one hidden layer with hidden units of several types and with various activation functions [5, 6, 10, 14, 15]. Other work provides rates of convergence in terms of the number of hidden neurons and properties of the functions to be approximated [1, 12]. For a survey see [16, 17], for example. Of course, the universal approximation capability is a key property for every machine learning mechanism used for approximation. Simple linear mappings, for example, do not possess this property and hence can only be applied to restricted domains. In order to establish SVMs as a universal mechanism, their universal approximation capability should be demonstrated. Note that they are similar to FNNs and hence the universal approximation capability can be expected. However, since they are trained in a different way and weight restrictions apply, a formal proof for their universal approximation capability is necessary.

One result in this direction is implicitely contained in [3]: SVMs with Gaussian kernels can approximate any finite number of data up to any desired accuracy if the parameters are chosen appropriately. We will show that SVMs with standard kernels including Gaussian, polynomial, and several dot product kernels can approximate any measurable or continuous function up to any desired accuracy. For this purpose, we will first introduce SVMs and the notation of approximation. Afterwards, we will transfer approximation results from FNNs to SVMs. Finally, we will show that a degree $p-1$ is sufficient for approximating p points with polynomial SVMs.

2. Support vector machines

Assume $(\vec{x}^1, y_1), \ldots, (\vec{x}^m, y_m) \in \mathbb{R}^n \times \{-1, 1\}$ are binary training patterns. If used for binary classification, the SVM consists of a simple linear classifier together with a fixed prior nonlinear mapping of the data into a high dimensional vector space such that data become linearly separable; *i.e.*, the SVM implements a mapping

$$\vec{x} \mapsto \mathrm{H}(\vec{w} \cdot \Phi(\vec{x}) + b)\,, \tag{1}$$

where $\Phi : \mathbb{R}^n \to X$ is a fixed nonlinear mapping, X is some possibly infinite dimensional vector space with dot product '$\cdot$', $\vec{w} \in X$ is the weight vector of the linear classifier, and b its bias. $\mathrm{H} : \mathbb{R} \to \{-1, 1\}$ is the Heaviside function with $\mathrm{H}(x) = 1$ iff $x \geq 0$. The generalization error of arbitrary linear classifiers scales with the dimensionality of the input space. Hence the separating hyperplane with maximum margin with respect to data is chosen such that good generalization is guaranteed. Then training can be formulated as a constraint optimization problem:

$$\min \frac{1}{2}|\vec{w}|^2 \text{ s.t. } y_i(\vec{w} \cdot \Phi(\vec{x}^i) + b) \geq 1\,. \tag{2}$$

For the sake of efficiency, only mappings Φ which can be kernelized are considered, *i.e.*, there exists a so-called kernel $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ such that the dot product reads as

$$\Phi(\vec{x}) \cdot \Phi(\vec{y}) = k(\vec{x}, \vec{y}) \tag{3}$$

for all vectors $\vec{x}$, $\vec{y}$. Using the Kuhn-Tucker conditions and the dual optimization problem yields a formulation for (2) which can be solved efficiently and which allows to substitute Φ by k:

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j k(\vec{x}^i, \vec{x}^j) \alpha_i \alpha_j \text{ s.t. } \alpha_i \geq 0, \sum_i \alpha_i y_i = 0 \tag{4}$$

where the Lagrange multipliers α_i lead to the representation $\vec{w} = \sum_i y_i \alpha_i \Phi(\vec{x}^i)$ and $b = y_j - \sum_i y_i \alpha_i k(\vec{x}^j, \vec{x}^i)$ for each j such that $\alpha_j \neq 0$. Hence an alternative notation for the classifier is

$$\vec{x} \mapsto \mathrm{H}\left(\sum_i \alpha_i y_i k(\vec{x}^i, \vec{x}) + b\right) . \tag{5}$$

Popular choices for the kernel k are

- polynomial kernels $k_d(\vec{x}, \vec{y}) \mapsto (\vec{x} \cdot \vec{y} + 1)^d$ with fixed degree $d \in \mathbb{N}$; k_d corresponds to $\Phi_d : \mathbb{R}^n \to \mathbb{R}^{\binom{n+d}{d}}$ where the image of $(x_1, \ldots, x_n)$ has coefficients $c_{i_1,\ldots,i_n} x_1^{i_1} \ldots x_n^{i_n}$ with positive factors $c_{i_1,\ldots,i_n} = ((d!)/(i_1! \ldots i_n!(d - i_1 - \ldots - i_n)!))^{1/2}$ and $\sum_j i_j \leq d$. The nonlinear mapping is not unique, neither is the codomain X of possible nonlinear mappings associated to k_d.

- dot product kernels $k_s(\vec{x}, \vec{y}) \mapsto f(\kappa \vec{x} \cdot \vec{y} - \delta)$ for appropriate fixed constants κ and δ and functions f. f is often chosen as a sigmoidal function. However, note that not every function f yields a proper kernel; in particular, the popular choice $f = \tanh$ does not lead to a kernel. Conditions on f such that a kernel results have been presented in [20], for example.

- Gaussian kernels $k_g(\vec{x}, \vec{y}) \mapsto \exp(-|\vec{x} - \vec{y}|^2/(2\delta^2))$ for fixed δ.

If used for regression, *i.e.*, the training patterns $(\vec{x}^i, y_i)$ are contained in $\mathbb{R}^n \times \mathbb{R}$, the SVM implements the linear mapping

$$\vec{x} \mapsto \vec{w} \cdot \Phi(\vec{x}) + b\,. \tag{6}$$

Training can be formulated as the optimization problem

$$\min \frac{1}{2}|\vec{w}|^2 \text{ s.t. } y_i - (\vec{w} \cdot \Phi(\vec{x}) + b) \leq \epsilon, (\vec{w} \cdot \Phi(\vec{x}) + b) - y_i \leq \epsilon \tag{7}$$

for some fixed accuracy $\epsilon > 0$. This formulation corresponds to the so-called ϵ-insensitive loss, where deviations of the desired value which are smaller than ϵ are neglected. Again, the dual problem and the Kuhn-Tucker conditions allow a notation as an optimization problem in Lagrange variables α_i and α_i^* which can be solved efficiently and which depends on k instead of Φ. It leads to a representation $\vec{w} = \sum_i (\alpha_i - \alpha_i^*)\Phi(\vec{x}^i)$ and hence mapping (6) becomes

$$\vec{x} \mapsto \sum_i (\alpha_i - \alpha_i^*) k(\vec{x}^i, \vec{x}) + b\,. \tag{8}$$

Note that mappings (5) and (8) are very similar to mappings implemented by standard multilayer feedforward networks with one hidden layer (FNNs) or radial basis function networks (RBFNNs), respectively. A FNN with activation function $\sigma : I\!R \to I\!R$ and linear output unit computes a mapping

$$\vec{x} \to \sum_{i=1}^{N} W_i \sigma(\vec{w}^i \cdot \vec{x} + d_i) + D\,, \tag{9}$$

where $N \in I\!N$ denotes the number of hidden nodes, $W_i \in I\!R$, $\vec{w}^i \in I\!R^n$ are the weights, and d_i, $D \in I\!R$ the biases which may be chosen arbitrarily. They are most often trained such that the empirical error on a given training set is minimal. If used for classification, the function H is added to the output in (9). A popular choice for the activation function σ is a sigmoidal function. Note that the only difference to an SVM (8) with dot product kernel consists in the fact that d_i is substituted by a fixed bias $-\delta$ in the SVM, the weights $\vec{w}^i$ come from training points $\vec{x}^i$, and the weights W_i and bias D cannot be chosen arbitrarily but are the solutions of an optimization problem, *i.e.*, they correspond to the mapping with optimum margin with respect to the given training patterns.

An RBFNN with activation function $\sigma : I\!R^n \to I\!R$ and linear output computes a mapping

$$\vec{x} \mapsto \sum_{i=1}^{N} W_i \sigma((\vec{x} - \vec{w}^i)/d_i) + D \tag{10}$$

with widths d_i, bias D, and weights $W_i \in I\!R$ and $\vec{w}^i \in I\!R^n$, $N \in I\!N$ denotes the number of hidden nodes. The function σ is often a radially symmetric function based on the exponential function and hence RBFNNs are very similar to SVMs with Gaussian kernel except for the fact that not every possible weight setting can occur in SVMs.

3. Approximation capability

Denote by $\mathcal{F}$ a function class $I\!R^n \to I\!R$ which is used for regression. We say that $\mathcal{F}$ possesses the universal approximation capability if any continuous function can be approximated on compacta in the maximum norm with some function in $\mathcal{F}$. That means for any compact set $C \subset I\!R^n$, any continuous function $g : C \to I\!R$, and any $\epsilon > 0$ some $f \in \mathcal{F}$ can be found such that $|f(\vec{x}) - g(\vec{x})| \leq \epsilon$ for all $\vec{x} \in C$. Assumed $\mathcal{F}$ is used for classification only, *i.e.*, the function's images are contained in $\{-1, 1\}$, then we say that $\mathcal{F}$ possesses the universal approximation capability with respect to a probability measure P on $I\!R^n$ if any measurable function with values in $\{-1, 1\}$ can be approximated arbitrarily well in probability. That means for any measurable function $g : I\!R^n \to \{-1, 1\}$ and $\delta > 0$ some $f \in \mathcal{F}$ exists such that $P(\{\vec{x} \mid f(\vec{x}) \neq g(\vec{x})\}) < \delta$.

There exists a couple of results concerning the universal approximation capability of neural networks. Denote by $\mathcal{FNN}(\sigma, \mathcal{D})$ the class of functions which can be computed by a FNN of form (9) with linear output, activation function $\sigma : I\!R \to I\!R$, and biases $d_i \in \mathcal{D} \subset I\!R$. Then it is shown in [9](Theorem 3) that $\mathcal{FNN}(\sigma, \mathcal{D})$ possesses the universal approximation capability if σ is some function which is analytic and nonpolynomial on a nondegenerate interval, and $\mathcal{D}$ contains a value such that all derivatives of σ are nonvanishing. Assumed we substituted in the above function class the linear output neuron by a neuron with the Heaviside activation function, then the resulting class $\mathcal{FNN}^{\mathrm{H}}(\sigma, \mathcal{D})$ possesses the universal approximation capability with respect to any probability P for classification problems. This is an immediate corollary of [9](Theorem 3): For any probability measure P, any measurable function $g : I\!R^n \to \{-1, 1\}$, and any $\delta > 0$ a continuous function $f_1 : I\!R^n \to I\!R$ exists with $P(\{\vec{x} \mid |g(\vec{x}) - f_1(\vec{x})| > 1/3\} < \delta/2$. We find a compact subset C of $I\!R^n$ such that $P(C) \geq 1 - \delta/2$. f_1 can be approximated up to accuracy $1/3$ on C with a function f_2 computed by a feedforward network with linear output. Then $H \circ f_2$ approximates g in probability with parameter δ.

Denote by $\mathcal{RBF}(\sigma, \mathcal{D})$ the class of functions which can be computed by an RBFNN of form (10) with linear output, activation function $\sigma : I\!R^n \to I\!R$, and widths $d_i \in \mathcal{D} \subset I\!R$. Then $\mathcal{RBF}(\sigma, \mathcal{D})$ fulfills the universal approximation capability if σ is an integrable, bounded, and continuous function such that $\int_{I\!R^n} \sigma(\vec{x}) d\vec{x} \neq 0$, and $\mathcal{D} \neq \emptyset$ [14](Theorem 2). If used for classification, *i.e.*, the functions are combined with the Heaviside function H yielding the function class $\mathcal{RBF}^{\mathrm{H}}(\sigma, \mathcal{D})$, the universal approximation capability holds under the weaker conditions that σ is integrable, $\int_{I\!R^n} \sigma(\vec{x}) d\vec{x} \neq 0$, and $\mathcal{D} \neq \emptyset$ [15](Theorem 1).

4. Approximation capability of support vector machines

We are interested in the universal approximation capability of functions given by an SVM. We call any mapping $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ a kernel if a Hilbert space X and a mapping $\Phi : \mathbb{R}^n \to X$ exist such that equation (3) holds. Denote by $\mathcal{SVM}(k)$ the class of functions which are computed by an SVM with kernel k which has been trained for regression with the ϵ-insensitive loss for some $\epsilon > 0$ on some training set, *i.e.*, any function of the form (6) such that some $\epsilon > 0$ and points $(\vec{x}^i, y_i)$ exist such that the function is a solution of (7). Denote by $\mathcal{SVM}^{\mathrm{H}}(k)$ the class of functions computed by an SVM with kernel k trained for classification on some training set, *i.e.*, any function of the form (1) such that points $(\vec{x}^i, y_i)$ exist such that the function is a solution of (2). As already mentioned, popular kernels allow a representation of the SVM as a standard FNN or RBFNN. Hence the universal approximation capability would follow immediately from the above mentioned universal approximation capability of FNNs or RBFNNs, respectively, if the Lagrange multipliers α_i and α_i^* and the bias b could be chosen arbitrarily. However, they correspond to the hyperplane with maximum margin with respect to the actual training set. Denote by $\mathcal{SVM}_g(k)$ the class of functions which are of the form (8) with kernel k and arbitrary parameters α_i, α_i^*, $\vec{x}^i$, and b. Denote by $\mathcal{SVM}_g^{\mathrm{H}}(k)$ the class of functions which can be denoted as (5) with arbitrary α_i, $\vec{x}^i$, and b.

The following observation is immediate:

Observation 1. Assume $\{(\vec{x}^i, y_i) \,|\, i = 1, \ldots, m\}$ is a training set for an SVM. Assume some vector $\vec{w}$ and bias b exist such that the constraints of optimization problem (2) or (7), respectively, are fulfilled. Then a solution of (2) or (7), respectively, exists and for every solution $\vec{w}^0$, b_0 the inequality $|\vec{w}^0| \leq |\vec{w}|$ holds.

This simple observation allows us to substitute the classes $\mathcal{SVM}(k)$ and $\mathcal{SVM}^{\mathrm{H}}(k)$, respectively, by the more general classes $\mathcal{SVM}_g(k)$ and $\mathcal{SVM}_g^{\mathrm{H}}(k)$, respectively, where the Lagrange multipliers and biases can be chosen arbitrarily:

Lemma 1. Assume $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a measurable kernel. If $\mathcal{SVM}_g^{\mathrm{H}}(k)$ possesses the universal approximation capability then so does $\mathcal{SVM}^{\mathrm{H}}(k)$.

Proof. Assume a measurable function $g : \mathbb{R}^n \to \mathbb{R}$, a probability measure P on $\mathbb{R}^n$, and some $\delta > 0$ are given. Assume $\mathcal{SVM}_g^{\mathrm{H}}(k)$ possesses the universal approximation capability. Then some function

$f_0 \in \mathcal{SVM}_g^{\mathrm{H}}(k)$ can be found with $P(\{\vec{x}\,|\,g(\vec{x}) \neq f_0(\vec{x})\}) < \delta/3$. Denote its weight vector by $\vec{w}^0$ which is a linear combination of points in the image of $I\!R^n$ under Φ. Choose $R > 0$ such that $P(\Phi^{-1}(B_R(\vec{0}))) \geq 1 - \delta/3$ where $B_R(\vec{0})$ is the closed ball with radius R centered in the origin in X. Note that $\Phi^{-1}(B_R(\vec{0})) = \{\vec{x} \mid k(\vec{x},\vec{x}) \leq R\}$ is measurable. Denote $C = \Phi^{-1}(B_R(\vec{0})) \cap \{\vec{x} \mid g(\vec{x}) = f_0(\vec{x})\}$. Note that $P(C) > 1 - 2\delta/3$. Denote by $P|_C$ the restriction of P to C. Assume a support vector machine is trained on m points $(\vec{x}^i, f_0(\vec{x}^i))$ where the $\vec{x}^i$ are chosen independently and identically distributed according to $P|_C$ and m will be defined below. Denote the weight vector of the SVM by $\vec{w}^1$, the bias by b_1, and the corresponding function by f_1. Because of Observation 1, we find $|\vec{w}^1| \leq |\vec{w}^0|$. Due to [18](5.5)

$$P|_C(\{\vec{x} \mid f_1(\vec{x}) \neq f_0(\vec{x})\}) \leq \frac{2}{m}\left(c \log \frac{8em}{c} \log(32\,m) + \log \frac{8em}{0.5}\right)$$

with confidence 0.5 where $c = 577\,|\vec{w}^1|^2 \max\{R, |b_1|\}^2$ and m denotes the number of points used for training. [1] Since $|\vec{w}^1|$ can be uniformly bounded by $|\vec{w}^0|$ and $|b_1| = |y_j - \sum_i y_i\alpha_i k(\vec{x}^i,\vec{x}^j)| \leq 1 + |\vec{w}^0|R$ for $\alpha_j \neq 0$ we can choose m such that the right hand side is smaller than $\delta/3$. Since g and f_0 coincide on C there exists a function f_1 such that $P|_C(\{\vec{x} \in C\,|\,f_1(\vec{x}) \neq g(\vec{x})\}) \leq \delta/3$. Hence there exists a support vector machine which computes $f_1 \in \mathcal{SVM}^{\mathrm{H}}(k)$ which coincides with g up to a set of probability δ on $I\!R^n$ with respect to P.

In a similar way, the corresponding result for SVMs used for regression can be obtained:

Lemma 2. Assume $k : I\!R^n \times I\!R^n \to I\!R$ is a continuous kernel. Then $\mathcal{SVM}(k)$ possesses the universal approximation capability if $\mathcal{SVM}_g(k)$ does.

Proof. Because of $|\Phi(\vec{x}) - \Phi(\vec{y})|^2 \leq |k(\vec{x},\vec{x}) - k(\vec{x},\vec{y})| + |k(\vec{y},\vec{y}) - k(\vec{x},\vec{y})|$, Φ is continuous. Assume $g : I\!R^n \to I\!R$ is continuous, $C \subset I\!R^n$ is a compact subset, and $\epsilon > 0$. If $\mathcal{SVM}_g(k)$ possesses the universal approximation capability then some $f_0 \in \mathcal{SVM}_g(k)$ can be found such that $|f_0(\vec{x}) - g(\vec{x})| < \epsilon/4$ for all $\vec{x} \in C$. Denote the weight vector and bias of f_0 by $\vec{w}^0$ and b_0, respectively. Since Φ is uniformly continuous on the compact set C, we find $\delta > 0$ such that $|\Phi(\vec{x}) - \Phi(\vec{y})| \leq \epsilon/(8\,|\vec{w}^0|)$

[1] Note that this bound on the generalization ability of SVMs is formulated in [18](5.5) only for finite dimensional real vector spaces since it is based on Vapnik's bound on the so-called fat-shattering dimension of large margin classifiers [21]. However, Gurvits provides the same bound on the fat-shattering dimension in the case of Hilbert spaces in [7], hence the above estimation can be applied in our case, too.

for all $\vec{x}$, $\vec{y} \in C$ with $|\vec{x} - \vec{y}| \leq \delta$. Choose points $\vec{x}^i$ in C such that the union of the sets $B_\delta(\vec{x}^i)$ covers C, where $B_\delta(\vec{x}^i)$ is the closed ball of radius δ centered in $\vec{x}^i$. Train an SVM on the points $(\vec{x}^i, g(\vec{x}^i))$ with the $\epsilon/4$-sensitive loss. Because the function f_0 fulfills the constraints in (7) there exists a solution $\vec{w}^1$, b_1 which yields an SVM with function f_1 such that $|\vec{w}^1| \leq |\vec{w}^0|$ due to Observation 1. $f_1 \in \mathcal{SVM}(k)$ approximates g on C with parameter ϵ: For any $\vec{x} \in C$ a training point $\vec{x}^1 \in C$ can be found such that $|\vec{x} - \vec{x}^1| \leq \delta$. Therefore

$$
\begin{aligned}
& |g(\vec{x}) - f_1(\vec{x})| \\
\leq\ & |g(\vec{x}) - f_0(\vec{x})| + |f_0(\vec{x}) - f_0(\vec{x}^1)| + |f_0(\vec{x}^1) - g(\vec{x}^1)| \\
& \qquad + |g(\vec{x}^1) - f_1(\vec{x}^1)| + |f_1(\vec{x}^1) - f_1(\vec{x})| \\
<\ & \epsilon/4 + |\vec{w}^0||\Phi(\vec{x}) - \Phi(\vec{x}^1)| + \epsilon/4 + \epsilon/4 + |\vec{w}^1||\Phi(\vec{x}) - \Phi(\vec{x}^1)| \\
\leq\ & \epsilon/4 + \epsilon/8 + \epsilon/4 + \epsilon/4 + \epsilon/8 = \epsilon
\end{aligned}
$$

Since $\mathcal{SVM}_g(k)$ and $\mathcal{SVM}_g^{\mathrm{H}}(k)$ coincide with standard networks for popular kernels k, the universal approximation capability of SVMs can now be established:

Theorem 1. Assume k is a measurable kernel of the form $k(\vec{x}, \vec{y}) = \sigma(\vec{x} \cdot \vec{y} + b)$ where $\sigma : \mathbb{R} \to \mathbb{R}$ is nonpolynomial and analytic on an open interval I and $b \in I$ such that all derivatives of σ in b do not vanish. Then $\mathcal{SVM}^{\mathrm{H}}(k)$ possesses the universal approximation capability. If k is in addition continuous, $\mathcal{SVM}(k)$ possesses the universal approximation capability, too.

Proof. This result follows immediately from Lemmas 1 and 2 because the classes $\mathcal{SVM}_g(k)$ or $\mathcal{SVM}_g^{\mathrm{H}}(k)$, respectively, coincide with $\mathcal{FNN}(\sigma, \{b\})$ and $\mathcal{FNN}^{\mathrm{H}}(\sigma, \{b\})$, respectively. The latter possess the universal approximation capability as shown in [9](Theorem 3).

Theorem 2. Assume k is a kernel of the form $k(\vec{x}, \vec{y}) = \sigma((\vec{x} - \vec{w})/d)$ where $\sigma : \mathbb{R}^n \to \mathbb{R}$ is an integrable function with $\int_{\mathbb{R}^n} \sigma(\vec{x}) d\vec{x} \neq 0$. Then $\mathcal{SVM}^{\mathrm{H}}(k)$ possesses the universal approximation capability. If in addition σ is a bounded and continuous function, $\mathcal{SVM}(k)$ possesses the universal approximation capability, too.

Proof. Again, this follows from Lemmas 1 and 2 and the universality of $\mathcal{SVM}_g(k)$ or $\mathcal{SVM}_g^{\mathrm{H}}(k)$, respectively, which coincide with $\mathcal{RBF}(\sigma, \{b\})$ or $\mathcal{RBF}^{\mathrm{H}}(\sigma, \{b\})$, respectively. The universal approximation capability of RBFNNs has been proved in [15](Theorem 1) and [14](Theorem 2).

Note that Theorems 1 and 2 include Gaussian and dot product kernels with appropriate function as a special case. SVMs with polynomial kernels with some fixed degree d are obviously not universal since they can implement at most polynomials of degree d. However, if d is not limited, the universality follows as well.

Theorem 3. $\bigcup_{d=1}^{\infty} \mathcal{SVM}(k_d)$ and $\bigcup_{d=1}^{\infty} \mathcal{SVM}^{\mathrm{H}}(k_d)$ possess the universal approximation capability, where k_d is the kernel $k_d(\vec{x},\vec{y}) = (\vec{x}\cdot\vec{y}+1)^d$.

Proof. It follows along the same lines as Lemmas 1 and 2 that it suffices to prove the universal approximation capability of $\bigcup_{d=1}^{\infty} \mathcal{SVM}_g(k_d)$ and $\bigcup_{d=1}^{\infty} \mathcal{SVM}_g^{\mathrm{H}}(k_d)$, respectively. The argumentation at the end of Section 3 shows that we can restrict to $\bigcup_{d=1}^{\infty} \mathcal{SVM}_g(k_d)$, *i.e.*, the universality for classification follows from the universality for regression. Assume $g : I\!R^n \to I\!R$ is a continuous mapping, $C \subset I\!R^n$ a compact set, and $\epsilon > 0$. It is well known that g can be approximated on C up to ϵ with a polynomial p_0. Assume the degree of p_0 is d_0. It will be shown that p_0 can be implemented precisely with a function in $\mathcal{SVM}_g(k_{d_0})$: Mappings in $\mathcal{SVM}_g(k_{d_0})$ have the form

$$\vec{x} \mapsto \sum_i \alpha_i(\vec{x}^i\cdot\vec{x}+1)^{d_0} + b$$

for parameters α_i, $b \in I\!R$ and points $\vec{x}^i = (x_1^i,\dots,x_n^i) \in I\!R^n$. We can write $(\vec{x}^i\cdot\vec{x}+1)^{d_0} = \sum_{i_1+\ldots+i_n\le d_0}(c_{i_1\ldots i_n})^2(x_1^i)^{i_1}\ldots(x_n^i)^{i_n}(x_1)^{i_1}\ldots(x_n)^{i_n}$ where $c_{i_1\ldots i_n} = ((d!)/(i_1!\ldots i_n!(d-i_1-\ldots-i_n)!))^{1/2}$ are nonzero constants. The monomials $(x_1)^{i_1}\ldots(x_n)^{i_n}$ for $i_1+\ldots+i_n \le d_0$ are linearly independent and span the vector space of all polynomials of degree at most d_0 in n variables. Denote $p = \binom{n+d_0}{d_0}$. Denote by $\mathrm{pol}_0,\ldots,\mathrm{pol}_p$ all monomials of degree at most d_0 of the form $(c_{i_1,\ldots,i_n})^2(x_1)^{i_1}\ldots(x_n)^{i_n}$. Since these monomials are linearly independent, the image of the mapping $\vec{x}\in\mathbb{R}^p \mapsto (\mathrm{pol}_1(\vec{x}),\ldots,\mathrm{pol}_p(\vec{x}))$ cannot be contained in a proper linear subspace of $\mathbb{R}^p$. Hence we can find p points $\vec{x}^i \in I\!R^n$ such that the matrix

$$\begin{pmatrix} \mathrm{pol}_1(\vec{x}^1) & \cdots & \mathrm{pol}_p(\vec{x}^1) \\ \vdots & & \vdots \\ \mathrm{pol}_1(\vec{x}^p) & \cdots & \mathrm{pol}_p(\vec{x}^p) \end{pmatrix}$$

has full rank. Since the rows are linearly independent, the points $\vec{x}^i$, an appropriate choice of α_i, and $b = 0$ allow to obtain $p_0 = \sum_i \alpha_i(\vec{x}^i\cdot\vec{x}+1)^{d_0}+b$.

Hence SVMs with polynomial kernel are universal approximators, too, assumed the degree of the kernel is not limited. Of particular interest for practical implementations is the question which degree is sufficient for the approximation of a finite number of points, *i.e.*, which degree is sufficient for a concrete training setting.

Theorem 4. Assumed a training set of size p with points $(\vec{x}^i, y_i)$ is given with pairwise different $\vec{x}^i$. Then an SVM with polynomial kernel with degree $p-1$ which is trained on these points can approximate the outputs y_i to any accuracy $\epsilon > 0$.

Proof. Because of Observation 1 it is sufficient to show that the images $\Phi_p(\vec{x}^i)$ are linearly independent, where Φ_p is the mapping with coefficients $c_{i_1,\ldots,i_n}(x_1)^{i_1}\cdots(x_n)^{i_n}$, where $i_1+\ldots+i_n \leq p-1$, the positive factors $c_{i_1,\ldots,i_n}$ are as above, and $\Phi_p(\vec{x})\cdot\Phi_p(\vec{y}) = (\vec{x}\cdot\vec{y}+1)^p$. Since the values $c_{i_1,\ldots,i_n}$ are positive, it suffices to prove that the matrix

$$M = \begin{pmatrix} \cdots & \left[(x_1^1)^{i_1}\ldots(x_n^1)^{i_n}\right] & \cdots \\ & \vdots & \\ \cdots & \left[(x_1^p)^{i_1}\ldots(x_n^p)^{i_n}\right] & \cdots \end{pmatrix}$$

has full rank where each column comprises one of the monomials of degree at most $p-1$ in n variables of the form $(x_1)^{i_1}\ldots(x_n)^{i_n}$ evaluated at $\vec{x}^1, \ldots, \vec{x}^p$. Since $\vec{x}^i \neq \vec{x}^j$ for all $i \neq j$ we find a linear mapping $h : I\!R^n \to I\!R$ such that the images $h(\vec{x}^i)$ are pairwise different. We can find a representation $h(x_1,\ldots,x_n) = \sum_{i=1}^n \alpha_i x_i$ for appropriate $\alpha_i \in I\!R$, therefore the terms $(h(\vec{x}^i))^j$ for $j \leq p-1$ constitute polynomials of degree at most $p-1$ in n variables. Since M comprises all monomials of degree at most $p-1$, the columns of the following matrix can be obtained via linear combinations of the columns of M:

$$\tilde{M} = \begin{pmatrix} 1 & (h(\vec{x}^1)) & \cdots & (h(\vec{x}^1))^{p-1} \\ \vdots & \vdots & & \vdots \\ 1 & (h(\vec{x}^p)) & \cdots & (h(\vec{x}^p))^{p-1} \end{pmatrix}$$

$\tilde{M}$ constitutes a Vandermonde matrix with full rank, since the values $h(\vec{x}^i)$ are pairwise different. Hence the rank of $\tilde{M}$ and M is p.

5. Conclusions

Note that Theorems 1, 2, and 3 include the universal approximation capability of SVMs with various dot product, Gaussian, and polynomial kernels. Hence SVMs possess the key properties of universal tools for machine learning: They provide valid generalization, they are universal approximators, and they are efficiently trainable. Moreover, it has been shown that every training set of size p can be approximated with polynomial kernels of degree $p-1$. A result analogous to Theorem 4 is implicitely contained in [3] for Gaussian kernels: For any finite training set with patterns $(\vec{x}^i, y_i)$ an appropriate width δ can be found such that an SVM with Gaussian kernel $k_g(\vec{x}, \vec{y}) = \exp(-|\vec{x} - \vec{y}|^2/(2\delta^2))$ approximates the outputs y_i up to any desired accuracy if trained on the patterns $(\vec{x}^i, y_i)$. This is due to the fact that small widths δ yield terms $\exp(-|\vec{x}^i - \vec{x}^j|^2/(2\delta^2))$ which approximate 1 iff $\vec{x}^i = \vec{x}^j$ and 0, otherwise. Hence one can easily obtain a linear combination of these terms such that the desired outputs on the training points $\vec{x}^i$ arise.

However, a couple of questions are still open: Note, that Theorems 1 and 2 do not imply that an SVM which has been trained on a particular set of points $(\vec{x}^i, y_i)$ yields a small training error. Due to the universal approximation capability we know that some SVM, say it is given by $\vec{x} \mapsto \sum \alpha_j k(\vec{w}^j, \vec{x}) + b$, exists which could approximate the points up to any accuracy. However, the vectors $\vec{w}^j$ might be different from the training points $\vec{x}^i$, and the vector $\sum \alpha_j \Phi(\vec{w}^j)$ might not be contained in the span of the images $\Phi(\vec{x}^i)$ at all. As mentioned above, the question has been answered for Gaussian and polynomial kernels, it is still open for dot product kernels.

Another interesting direction of research refers to the rate of approximation. Note that an approximation rate of $1/\sqrt{n}$, n being the number of hidden neurons, could be established for FNNs [1]. Rates for SVMs would be very interesting, too. Note that the number of hidden neurons and the interior weights are determined by the training patterns in the SVM. Hence results on the approximation rate should take into account that the interior weights correspond to training points chosen independent and identically distributed according to some probability.

Note that the universal approximation capability of SVMs does not imply that every function can be approximated with a large margin, *i.e.*, small weight vector $\vec{w}$. It has been shown in [2] that a large number of functions cannot be embedded such that a large margin arises. Hence another topic of research is which kernels are appropriate for which functions to be approximated such that a large margin can be obtained.

References

1. A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39 (1993), 930-945.

2. S. Ben-David, N. Eiron, and H. U. Simon. *Limitations of Learning via Embedding in Euclidean Half Spaces.* NeuroCOLT Technical Report 2001-086, 2001.
3. C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) (1998), 121-167.
4. C. Cortes and V. Vapnik. Support vector network. *Machine Learning*, 20 (1995), 1-20.
5. G. Cybenko. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2 (1989), 303-314.
6. K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2 (1989), 183-192.
7. L. Gurvits. A note on a scale-sensitive dimension of linear bounded functionals in Banach spaces. In: M. Li and A. Maruoska (eds.), *Algorithmic Learning Theory ALT-97*, Springer, 1997, pp.352-363.
8. R. Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. In: *International Conference on Neural Networks*, vol. 3, IEEE, Washington DC, 1989, pp.11-14.
9. K. Hornik. Some new results on neural network approximation. *Neural Networks*, 6 (1993), 1069-1072.
10. K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2 (1989), 359-366.
11. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In: *Proceedings of the European Conference on Machine Learning*, Springer, Berlin, 1998, pp.137-142.
12. V. Kurkova, P. C. Kainen, and V. Kreinovich. Estimates of the number of hidden neurons and variation with respect to half-spaces. *Neural Networks*, 10 (1997), 1061-1068.
13. K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In: W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud (eds.) *Artificial Neural Networks – ICANN'97*, Springer, 1997, pp.999-1004.
14. J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2) (1991), 246-257.
15. J. Park and I. W. Sandberg. Approximation and radial-basis-function networks. *Neural Computation*, 5 (1993), 305-316.
16. A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, (1999), 143-195.
17. F. Scarselli and A. C. Tsoi. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, 11(1) (1998), 15-37.
18. J. Shawe-Taylor, P. L. Bartlett, R. Williamson, and M. Anthony. Structural risk minimization over data dependent hierarchies. Technical report, NeuroCOLT, 1996.
19. B. Schölkopf and A. J. Smola, *Learning with kernels.* The MIT Press, 2001.
20. A. Smola, *Support Vector Machines.* Tutorial at ICANN'01, Vienna, Austria, 2001.
21. V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition.* Nauka, Moscow, 1974.