1 Geometry and Invariance in Kernel Based Methods

Christopher J.C. Burges
Bell Laboratories, Lucent Technologies
101 Crawford's Corner Road, Holmdel, NJ 07733-3030, USA
burges@lucent.com
http://svm.research.bell-labs.com

To Appear In:
Advances in Kernel Methods - Support Vector Learning, Eds. B. Schölkopf, C. Burges, A. Smola, MIT Press, Cambridge, USA, 1998

We explore the questions of (1) how to describe the intrinsic geometry of the manifolds which occur naturally in methods, such as support vector machines (SVMs), in which the choice of kernel specifies a nonlinear mapping of one's data to a Hilbert space; and (2) how one can find kernels which are locally invariant under some given symmetry. The motivation for exploring the geometry of support vector methods is to gain a better intuitive understanding of the manifolds to which one's data is being mapped, and hence of the support vector method itself: we show, for example, that the Riemannian metric induced on the manifold by its embedding can be expressed in closed form in terms of the kernel. The motivation for looking for classes of kernels which instantiate local invariances is to find ways to incorporate known symmetries of the problem into the model selection (i.e. kernel selection) phase of the problem. A useful by-product of the geometry analysis is a necessary test which any proposed kernel must pass if it is to be a support vector kernel (i.e. a kernel which satisfies Mercer's positivity condition); as an example, we use this to show that the hyperbolic tangent kernel (for which the SVM is a two-layer neural network) violates Mercer's condition for various values of its parameters, a fact noted previously only experimentally. A basic result of the invariance analysis is that directly imposing a symmetry on the class of kernels effectively results in a preprocessing step, in which the preprocessed data lies in a space whose dimension is reduced by the number of generators of the symmetry group. Any desired kernels can then be used on the preprocessed data. We give a detailed example of vertical

translation invariance for pixel data, where the binning of the data into pixels has some interesting consequences. The paper comprises two parts: Part 1 studies the geometry of the kernel mapping, and Part 2 the incorporation of invariances by choice of kernel.

Part 1: The Geometry of the Kernel Mapping

1.1 Overview

A Support Vector Machine, whether for pattern classification, regression estimation, or operator inversion, uses a device called *kernel mapping* (Boser et al., 1992; Vapnik, 1995; Burges, 1998) to map the data to a Hilbert space $\mathcal{H}$ ("feature space") in which the problem becomes linear (i.e. an optimal separating hyperplane for the pattern recognition case, and a linear regression for the regression and operator inversion cases). If the original data lies in a d-dimensional space, the mapped data will lie in an at most d-dimensional surface $\mathcal{S}$ in $\mathcal{H}$. In Part 1 we explore the intrinsic geometry of these surfaces. Our main motivation is simply to gain a better intuitive understanding of the geometry underlying the support vector approach; however, some useful results will follow. We will show that the surface has an induced Riemannian metric which can be expressed solely in terms of the kernel, and that a given metric is generated by a class of kernels. We derive the volume element in the surface for dot product kernels, and show that positivity of the metric gives some simple necessary tests for whether a proposed kernel is indeed a support vector kernel (i.e. whether it satisfies Mercer's condition: see also Smola et al. (1998)). Note that the kernel mapping device is beginning to find applications beyond support vector machines (Schölkopf et al., 1998b,c); the work presented here applies to all such algorithms.

In the following, bold typeface will indicate vector or matrix quantities; normal typeface will be used for vector and matrix components and for scalars. Repeated indices are assumed summed.

1.2 The Kernel Mapping

We briefly remind the reader of the kernel mapping used by the support vector approach (Boser et al., 1992; Vapnik, 1995; Burges, 1998). Suppose one has data $\mathbf{x}_i \in \mathbf{R}^{d_L}, \quad i = 1, \cdots, l$ (we do not need to consider the labels $y_i \in \{\pm 1\}$ for the pattern recognition case here). For any symmetric, continuous function $K(\mathbf{x}, \mathbf{y})$ satisfying Mercer's condition, there exists a Hilbert space $\mathcal{H}$, a map $\Phi : \mathbf{R}^{d_L} \mapsto \mathcal{H}$, and positive numbers λ_n such that (Courant and Hilbert, 1953):

$$K(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{d_F} \lambda_n \Phi_n(\mathbf{x}) \Phi_n(\mathbf{y}) \tag{1.1}$$

where d_F is the dimension of $\mathcal{H}$ (note that one can add points to give a complete space if necessary). Mercer's condition requires that

Mercer's Condition

$$\int_{\mathcal{C}} K(\mathbf{x},\mathbf{y})g(\mathbf{x})g(\mathbf{y})d\mathbf{x}d\mathbf{y} \geq 0, \tag{1.2}$$

for any square integrable function $g(\mathbf{x})$, and where $\mathcal{C}$ is some compact subset of $\mathbf{R}^{d_L}$. For the purposes of this work, we absorb the λ_n into the definition of Φ, so that Eq. (1.1) becomes a dot product (in fact, an expansion (1.1) can still be found if a finite number of the λ_n are negative (Courant and Hilbert, 1953)). Below, following Stewart (1978), we will call a continuous symmetric function which satisfies Eq. (1.2) a *positive semidefinite kernel.*

Positive Semidefinite Kernel

1.2.1 Smoothness Assumptions

We make explicit the assumptions we will need. Label the subset of $\mathbf{R}^{d_L}$ on which the data lives $\mathcal{L}$ (we assume that the data is continuous). Given a positive semidefinite kernel with eigenfunctions $\Phi_n, \quad n = 1, \cdots, d_F$, we assume that some subset $\mathcal{L}_s \in \mathcal{L}$ exists on which the Φ_n are C^3 (i.e. have up to third derivatives defined and continuous), and on which the rank of the mapping Φ is some fixed number k (we usually encounter the case $k = d_L$). We will use $\mathcal{S}$ to denote the image of $\mathcal{L}_s$ under Φ. Thus $\mathcal{S}$ is a k-dimensional surface in $\mathcal{H}$.

Differentiable Manifold

It is instructive to consider under what conditions $\mathcal{S}$ will be a manifold, and under what conditions a differentiable (C^∞) manifold. In order for $\mathcal{S}$ to be a manifold of dimension k, it must be (i) Hausdorff[1], (ii) locally Euclidean of dimension k, and (iii) have a countable basis of open sets (Boothby, 1986). Condition (i) follows automatically (every Hilbert space is a normal space, hence a metric space, hence Hausdorff (Kolmogorov and S.V.Fomin, 1970)). Condition (iii) certainly follows if $\mathcal{H}$ is separable, since a Hilbert space has a countable basis of open sets if and only if it is separable, and $\mathcal{S}$, if itself an open set, can inherit the countable basis from $\mathcal{H}$. Condition (ii) will hold whenever the map Φ is of some fixed rank k everywhere. In order for $\mathcal{S}$ to be a differentiable manifold, it is necessary and sufficient that all derivatives of the map Φ exist and are continuous, and that Φ be of rank k everywhere.

Note that the requirement that $\mathcal{S}$ be a differentiable manifold is stronger than we need. In general we need only consider C^3 manifolds, and we allow singularities. The C^3 condition will enable us to define the Riemannian curvature for the surface.

1.3 Measures of Distance on $\mathcal{S}$

There are three measures of distance on $\mathcal{S}$ that spring to mind. They are shown schematically in Figure 1.1. There, the embedded surface is represented by a curve,

[1] Recall that a space $\mathcal{S}$ is Hausdorff if and only if, for any pair of distinct points $p_1, p_2 \in \mathcal{S}$, two open sets $S_1, S_2 \in \mathcal{S}$ can be found such that $p_1 \in S_1,\ p_2 \in S_2,\ S_1 \cap S_2 = \emptyset$.

and two points in the surface by P_1 and P_2. In case I, one considers the intrinsic distance measured along the surface itself. This distance is the distance function generated by a Riemannian metric on $\mathcal{S}$. In case II, one considers the Euclidean distance measured between the two points in $\mathcal{H}$. In this case, the line joining P_1 and P_2 will in general leave the surface $\mathcal{S}$. In case III, one considers projections of the position vectors of P_1 and P_2 along some vector $\mathbf{w} \in \mathcal{H}$. This is an affine distance function (for example, distances can be positive or negative) and it is the distance measure used in the support vector expansion (i.e. the decision rule, for the pattern recognition case, or the approximation to the function, in the regression case).

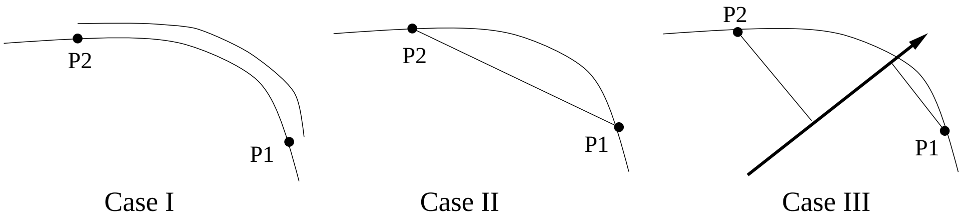

Figure 1.1 Measures of distance on $\mathcal{S}$.

Metric

Recall that in order for a finite real valued function $d(\mathbf{x},\mathbf{y})$, $\mathbf{x}$, $\mathbf{y} \in \mathbf{R}^{d_L}$ to earn the name "metric", it must satisfy the following conditions[2]:

$$d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x}) \geq 0 \tag{1.3}$$

$$d(\mathbf{x},\mathbf{y}) = 0 \iff \mathbf{x} = \mathbf{y} \tag{1.4}$$

$$d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{z}) \geq d(\mathbf{z},\mathbf{x}) \tag{1.5}$$

Riemannian Metric

A "Riemannian metric," on the other hand, is a symmetric, positive definite, bilinear form defined on a manifold, and if a manifold admits such a form, it is termed a "Riemannian manifold" (Boothby, 1986). A Riemannian metric defines a metric, such that the distance between two points $\mathbf{x}(t_0)$ and $\mathbf{x}(t_1)$, along a path lying in $\mathcal{S}$ and parameterized by $t \in [t_0, t_1]$, is given by the path integral below, where the expression under the square root in the integrand is the double contraction of the metric form with the tangent vector to the given path. Thus the $g_{\mu\nu}$ are the components (in a given coordinate system) of the metric tensor (Boothby, 1986).

$$\rho(t_0, t_1) = \int_{x(t_0)}^{x(t_1)} \left(g_{\mu\nu} \frac{\partial x^\mu}{\partial t} \frac{\partial x^\nu}{\partial t} \right)^{\frac{1}{2}} dt \tag{1.6}$$

Now consider the metric induced on $\mathcal{S}$ by the Euclidean metric on $\mathcal{H}$, i.e. case II above. This is not a Riemannian metric, since the distances cannot be expressed

[2] Note that Eqs. (1.3) in fact follow from Eqs. (1.4) and (1.5) (E.T.Copson, 1968).

in the form of Eq. (1.6) (suppose otherwise, and consider some path in $\mathcal{S}$ joining three points P_1, P_2 and P_3; then $d(P_1, P_2) + d(P_2, P_3)$ can in general be greater than $d(P_1, P_3)$, but Eq. (1.6) requires that the distance along the path from P_1 to P_2 plus that along the path from P_2 to P_3 equal the distance along the path from P_1 to P_3). Since in this work we are only interested in the intrinsic geometry of the surface $\mathcal{S}$, we will consider only case I.

1.4 From Kernel to Metric

For any positive kernel, and subset of the data, satisfying the assumptions listed above, the image $\mathcal{S}$ of the associated mapping will be a Riemannian manifold. We can easily find the induced Riemannian metric on $\mathcal{S}$. The line element on $\mathcal{S}$ can be written (here and below, Roman indices will run from 1 to d_F, and Greek from 1 to d_L, unless otherwise stated):

$$\begin{aligned} ds^2 &= g_{ab} d\Phi^a(\mathbf{x}) d\Phi^b(\mathbf{x}), \\ &= g_{\mu\nu} dx^\mu dx^\nu, \end{aligned} \tag{1.7}$$

where $g_{\mu\nu}$ is the induced metric, and the surface $\mathcal{S}$ is parameterized by the x_μ. Letting $d\mathbf{x}$ represent a small but finite displacement, we have

$$\begin{aligned} ds^2 &= \|\Phi(\mathbf{x} + d\mathbf{x}) - \Phi(\mathbf{x})\|^2 \\ &= K(\mathbf{x} + d\mathbf{x}, \mathbf{x} + d\mathbf{x}) - 2K(\mathbf{x}, \mathbf{x} + d\mathbf{x}) + K(\mathbf{x}, \mathbf{x}) \\ &= \left((1/2)\partial_{x_\mu}\partial_{x_\nu} K(\mathbf{x}, \mathbf{x}) - \partial_{y_\mu}\partial_{y_\nu} K(\mathbf{x}, \mathbf{y})\right)_{\mathbf{y}=\mathbf{x}} dx^\mu dx^\nu \\ &= g_{\mu\nu} dx^\mu dx^\nu \end{aligned}$$

Metric Tensor

Thus we can read off the components of the metric tensor:

$$g_{\mu\nu} = (1/2)\partial_{x_\mu}\partial_{x_\nu} K(\mathbf{x}, \mathbf{x}) - \{\partial_{y_\mu}\partial_{y_\nu} K(\mathbf{x}, \mathbf{y})\}_{\mathbf{y}=\mathbf{x}} \tag{1.8}$$

Let's illustrate this with some very simple examples.

Circle: Suppose Φ is the map from the line segment onto the circle of radius r: $\Phi : [0, 2\pi) \mapsto S^1$, i.e.

$$\Phi : \theta \mapsto \begin{pmatrix} r\cos\theta \\ r\sin\theta \end{pmatrix}, \quad r \quad \text{fixed} \tag{1.9}$$

Then

$$K(\theta, \theta') = r^2(\cos\theta\cos\theta' + \sin\theta\sin\theta') = r^2\cos(\theta - \theta') \tag{1.10}$$

and

$$ds^2 = \{(1/2)\partial_\theta^2 K(\theta, \theta) - \partial_{\theta'}\partial_{\theta'} K(\theta, \theta')\}_{\theta=\theta'} d\theta^2 \tag{1.11}$$

$$= r^2 d\theta^2 \tag{1.12}$$

2-Sphere: Here Φ maps $[0,\pi] \times [0, 2\pi) \mapsto S^2$:

$$\Phi : \{\theta, \psi\} \mapsto \begin{pmatrix} r\sin\theta\cos\psi \\ r\sin\theta\sin\psi \\ r\cos\theta \end{pmatrix} \tag{1.13}$$

Letting ξ be the vector with components θ, ψ,

$$K(\xi_1, \xi_2) = r^2 \sin\theta_1 \cos\psi_1 \sin\theta_2 \cos\psi_2 \tag{1.14}$$

$$+r^2 \sin\theta_1 \sin\psi_1 \sin\theta_2 \sin\psi_2 \tag{1.15}$$

$$+r^2 \cos\theta_1 \cos\theta_2 \tag{1.16}$$

which gives

$$g_{\mu\nu} = \begin{pmatrix} r^2 & 0 \\ 0 & r^2 \sin^2\theta \end{pmatrix} \tag{1.17}$$

A Fourier Sum: The Dirichlet kernel,

$$K(x_1, x_2) = \frac{\sin((N + \frac{1}{2})(x_1 - x_2))}{2\sin((x_1 - x_2)/2)}, \quad x_1, \ x_2 \in \mathbf{R}, \tag{1.18}$$

corresponds to a mapping Φ into a space $\mathcal{H}$ of dimension $2N+1$, where for any $\mathbf{a} \in \mathbf{R}^{2N+1}$, $\mathbf{a} \cdot \Phi(x)$ may be viewed as a Fourier expansion cut off after N terms, with coefficients $a_i, \ , i = 1, \ldots, 2N+1$ (Vapnik, 1995; Vapnik et al., 1997; Burges, 1998). As we shall see below, all kernels $K(\mathbf{x}, \mathbf{y})$ which take the form $K(\mathbf{x} - \mathbf{y})$ result in flat manifolds. The above kernel gives line element

$$ds^2 = \frac{1}{6} N(2N+1)(N+1) dx^2 \tag{1.19}$$

1.5 From Metric to Kernel

One might want to start with a chosen metric $g_{\mu\nu}$ on the data (for example, one that separates different classes according to some chosen criterion), and ask: is there a Mercer kernel for which the metric induced on $\mathcal{S}$, by its embedding in $\mathcal{H}$, is $g_{\mu\nu}$? By using such a kernel, for example, in a support vector machine, one would be explicitly controlling the intrinsic shape of the embedded surface $\mathcal{S}$. As we will see below, the answer is yes: any Riemannian manifold can be isometrically embedded in a Euclidean space (Nash, 1956). The above analysis shows that the corresponding kernel K must then be a solution to the differential equation (1.8). However, construction of such a kernel, given a metric, may not be straightforward. Although it is guaranteed that there is at least one positive symmetric (and continuous) kernel which satisfies (1.8), in general there will be solutions to (1.8) that are not positive, or that are not symmetric, or both. Note, however, that if one can find a positive symmetric solution to Eq. (1.8) (which will also, by construction, be continuous), then we know how to construct the embedding explicitly: by Mercer's theorem the expansion (1.1) exists, and the embedding is simply given by the eigenfunctions of the found solution, i.e. the Φ in Eq. (1.1).

1.6 Dot Product Kernels

One commonly used class of kernels are the dot product kernels (Vapnik, 1995; Burges and Schölkopf, 1997), where $K(\mathbf{x},\mathbf{y}) = K(\mathbf{x}\cdot\mathbf{y})$. Using Eq. (1.8) one finds a general expression for the corresponding Riemannian metric:

$$g_{\mu\nu} = \delta_{\mu\nu}K'(\|\mathbf{x}\|^2) + x_\mu x_\nu K''(\|\mathbf{x}\|^2), \tag{1.20}$$

where the prime denotes the derivative with respect to the argument $\|\mathbf{x}\|^2$. A simple example is given by the identity map, where $\Phi_\mu(\mathbf{x}) = x_\mu,\quad K(\mathbf{x},\mathbf{y}) = \mathbf{x}\cdot\mathbf{y},\quad g_{\mu\nu} = \delta_{\mu\nu}$.

Christoffel Symbols

Recall that the Christoffel symbols of the second kind are defined by

$$\Gamma^\alpha_{\beta\gamma} \equiv g^{\alpha\mu}\Gamma_{\beta\gamma\mu} = \frac{1}{2}g^{\alpha\mu}(\partial_\beta g_{\gamma\mu} - \partial_\mu g_{\beta\gamma} + \partial_\gamma g_{\mu\beta}) \tag{1.21}$$

where $g^{\alpha\mu}$ is the inverse of the matrix $g_{\alpha\mu}$ and ∂_α is shorthand for $\frac{\partial}{\partial x^\alpha}$. Note that the Christoffel symbols of the first kind have a particularly simple representation (for any positive semidefinite kernel):

$$\Gamma_{\alpha\beta\gamma} = \sum_{n=1}^{d_F}(\partial_\alpha\partial_\beta\Phi_n(\mathbf{x}))\partial_\gamma\Phi_n(\mathbf{x}) \tag{1.22}$$

The Riemannian metric given above has the form of a projection operator, and its contravariant components are therefore easily found:

$$g^{\mu\nu} = \frac{\delta_{\mu\nu}}{K'} - x_\mu x_\nu \frac{K''/K'}{K' + \|\mathbf{x}\|^2 K''} \tag{1.23}$$

Having the above closed form for $g^{\mu\nu}$ in terms of the kernel greatly facilitates the computation of several other quantities of interest. In the next Section we will use it to derive the curvature for homogeneous dot product kernels in arbitrary numbers of dimensions and for polynomials of arbitrary degree. We end this Section by noting that it immediately leads to a closed form for the Christoffel symbols of the second kind (for dot product kernels):

$$\Gamma^\rho_{\mu\nu} = \frac{K''}{K'}(x_\mu\delta_{\rho\nu} + x_\nu\delta_{\mu\rho}) + \frac{x_\mu x_\nu x_\rho}{K' + \|\mathbf{x}\|^2 K''}(K''' - 2(K'')^2/K') \tag{1.24}$$

1.6.1 The Curvature for Polynomial Maps

We can gain some insight into the nature of the corresponding surfaces by computing the curvature. We remind the reader that the intrinsic curvature is completely specified by the Riemann tensor (Dodson and Poston, 1991), with components:

Riemann Tensor

$$R_{\nu\alpha\beta}{}^{\mu} = \partial_\alpha \Gamma^\mu_{\beta\nu} - \partial_\beta \Gamma^\mu_{\alpha\nu} + \Gamma^\rho_{\alpha\nu}\Gamma^\mu_{\beta\rho} - \Gamma^\rho_{\beta\nu}\Gamma^\mu_{\alpha\rho} \tag{1.25}$$

For homogeneous polynomial kernels $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^p$ (for which Φ is a polynomial map), the metric is

$$g_{\mu\nu} = \delta_{\mu\nu} p (\|\mathbf{x}\|^2)^{p-1} + x_\mu x_\nu p(p-1)(\|\mathbf{x}\|^2)^{p-2} \tag{1.26}$$

and, using Eq. (1.24), we find

$$\Gamma^\rho_{\mu\nu} = \frac{p-1}{\|\mathbf{x}\|^2}\left(x_\mu \delta_{\rho\nu} + x_\nu \delta_{\mu\rho} - \frac{x_\mu x_\nu x_\rho}{\|\mathbf{x}\|^2}\right) \tag{1.27}$$

For $d_L = 2$, we find that these manifolds are flat, for all powers p. For $p = 2$ and $d_L = 2$, we can plot the surface, as shown below (Burges, 1998):

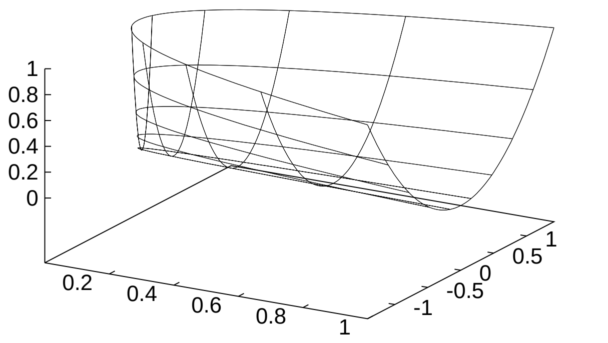

Figure 1.2 Image, in $\mathcal{H}$, of the square $[-1, 1] \times [-1, 1] \in \mathbf{R}^2$ under the mapping Φ.

We use Figure 1.2 to emphasize that we are dealing with the intrinsic curvature of the surface. A geometer living on this surface would conclude that the surface is flat because parallel transport of vectors around closed geodesic loops would not give any resulting displacement. Alternatively, one can bend a flat sheet of paper into the surface shown in Figure 1.2, without stretching it.

Ricci Tensor

Scalar Curvature

For $d_L \geq 3$, the manifold is not flat. Recalling that the Ricci tensor is defined by $R_{\mu\nu} = R_{\mu\alpha\nu}{}^{\alpha}$, and the scalar curvature by $R = R_\mu{}^\mu$, for general dimension d_L and power p, we have (note the singularity at $\mathbf{x} = 0$):

$$\begin{aligned} R_{\mu\nu\rho}{}^{\sigma} &= \frac{(p-1)}{\|\mathbf{x}\|^2}(\delta_{\rho\nu}\delta_{\sigma\mu} - \delta_{\rho\mu}\delta_{\sigma\nu}) \\ &- \frac{(p-1)}{\|\mathbf{x}\|^4}(x_\nu x_\rho \delta_{\mu\sigma} - x_\mu x_\rho \delta_{\sigma\nu} + x_\sigma x_\mu \delta_{\rho\nu} - x_\sigma x_\nu \delta_{\rho\mu}) \end{aligned} \tag{1.28}$$

$$R_{\mu\rho} = \frac{(p-1)(2-d_L)}{\|\mathbf{x}\|^2}\left(\delta_{\mu\rho} - \frac{x_\mu x_\rho}{\|\mathbf{x}\|^2}\right) \tag{1.29}$$

$$R = \frac{(p-1)(2-d_L)(d_L-1)}{p(\|\mathbf{x}\|^2)^p} \tag{1.30}$$

1.6.2 The Volume Element

The volume element of the mapped surface is of interest for several reasons. For example, one might expect poor generalization performance in the two-class pattern recognition case if some portion of the input space, containing data from both classes, gets mapped to a volume in $\mathcal{S}$ which is very small compared to the volume occupied by the rest of the mapped training data[3]. Secondly, given a probability density $p(\mathbf{x})$ for the data in input space, knowing the volume element, $dV = G(\mathbf{x})d\mathbf{x}$, immediately yields the density $\bar{p}(\mathbf{x})$ of the mapped data in $\mathcal{S}$ (assuming that the mapping is 1-1, and again treating the $\mathbf{x}$ in the latter case as a parameterization of the surface), since then $p(\mathbf{x})d\mathbf{x} = \bar{p}(\mathbf{x})G(\mathbf{x})d\mathbf{x}$.

For the case of dot product kernels we can explicitly compute the volume element $dV = \sqrt{(\det g_{\mu\nu})}dx_1 \ldots dx_{d_L}$. Defining the matrix (all quantities are functions of $\|\mathbf{x}\|^2$)

$$A_{\mu\nu} \equiv x_\mu x_\nu (K''/K')) \tag{1.31}$$

and using the identity $\det(1+A) = e^{Tr\ln(1+A)}$, we find

$$dV = (K')^{\frac{d_L}{2}}\sqrt{\left(1 + \|\mathbf{x}\|^2\frac{K''}{K'}\right)}dx_1 \ldots dx_{d_L} \tag{1.32}$$

1.7 Positivity

It is straightforward to check that the induced metric is in general positive definite: combining Eqs. (1.1) and (1.8) gives

$$g_{\mu\nu} = \sum_{n=1}^{d_F}(\partial_{x_\mu}\Phi_n(x))(\partial_{x_\nu}\Phi_n(x)) \tag{1.33}$$

[3] One might also expect poor generalization performance if some data from both classes is mapped to a subset of the manifold with high curvature.

so for any vector $\mathbf{v} \in \mathbf{R}^{d_L}$, $\mathbf{v}^T g \mathbf{v} = \sum_{n,\mu} (v_\mu \partial_{x_\mu} \Phi_n(x))^2 > 0$. (If $d_F = \infty$, we must also require that the the sum on the right hand side be uniformly convergent, and that the derivatives be continuous, in order to be able to take the derivatives underneath the summation sign).

We can use the fact that any Riemannian metric must be positive definite to derive conditions that any prospective kernel must satisfy. Using Eq. (1.20), we can easily find the eigenvectors $\mathbf{V}$ of the metric for the general dot products kernels discussed above:

$$\begin{aligned} g_{\mu\nu} V_\nu &= V_\mu K'(\|\mathbf{x}\|^2) + x_\mu (\mathbf{x} \cdot \mathbf{V}) K''(\|\mathbf{x}\|^2) \\ &= \lambda V_\mu \end{aligned} \tag{1.34}$$

Thus the eigenvectors are (i) all $\mathbf{V}$ orthogonal to $\mathbf{x}$ ($d_L - 1$ of them), with eigenvalues $K'(\|\mathbf{x}\|^2)$, and (ii) $\mathbf{V} = \mathbf{x}$, with eigenvalue $K' + \|\mathbf{x}\|^2 K''$. Hence we arrive at the following

Proposition 1.1
Three necessary conditions for a dot product kernel, $K(\mathbf{x}, \mathbf{y}) = K(\mathbf{x} \cdot \mathbf{y})$, to be a positive semidefinite kernel, are:

$$K(\|\mathbf{x}\|^2) \geq 0 \tag{1.35}$$

$$K'(\|\mathbf{x}\|^2) \geq 0 \tag{1.36}$$

$$K'(\|\mathbf{x}\|^2) + \|\mathbf{x}\|^2 K''(\|\mathbf{x}\|^2) \geq 0 \tag{1.37}$$

where the prime denotes derivative with respect to $\|\mathbf{x}\|^2$. The first condition follows directly from Eq. (1.2) (choose a square integrable $g(\mathbf{x})$ which is strongly peaked around some point $\mathbf{x}$, and which falls rapidly to zero elsewhere: see Courant and Hilbert (1953)), the last two from the requirement that the metric have positive eigenvalues.

The kernel $K(\mathbf{x}, \mathbf{y}) = \tanh(a\mathbf{x} \cdot \mathbf{y} + b)$ has been noted as generating a support vector machine which is equivalent to a particular two-layer neural network (Boser et al., 1992; Vapnik, 1995). However it is also known that this only holds for certain values of the parameters a, b: for others, it was noted experimentally that Mercer's condition is violated (Vapnik, 1995). Proposition 1.1 shows that a, b must satisfy

$$b \geq 0 \tag{1.38}$$

$$a \geq 0 \tag{1.39}$$

$$1 - 2a\|\mathbf{x}\|^2 \tanh(a\|\mathbf{x}\|^2 + b) \geq 0 \tag{1.40}$$

Eq. (1.38) follows from (1.35), with the assumption that $\|\mathbf{x}\|^2$ can be chosen to be arbitrarily small; Eq. (1.39) follows from (1.36). The proposition also shows immediately that some proposed kernels, such as $e^{-\mathbf{x} \cdot \mathbf{y}}$, are in fact not Mercer kernels. Note that the proposition gives necessary but not sufficient conditions. However Mercer's condition is often not easy to verify (Eq. (1.2) must hold for *any* square integrable g), so such tests can be very useful.

Clearly, conditions (1.36) and (1.37) must also follow from Mercer's condition. In fact condition (1.37) can be derived by choosing $g(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0) - \delta(\mathbf{x} - \mathbf{x}_0 - \epsilon)$ in Eq. (1.2) and letting ϵ approach zero (δ is the Dirac delta function)[4].

For kernels of the form $K(\mathbf{x}, \mathbf{y}) = K(||\mathbf{x} - \mathbf{y}||^2)$, requiring positivity of the metric gives the following proposition:

Proposition 1.2

Given a function K with first derivatives defined and continuous, two necessary conditions for $K(||\mathbf{x} - \mathbf{y}||^2)$ to be a positive definite kernel are

$$K(0) > 0 \tag{1.41}$$

$$K'(0) < 0 \tag{1.42}$$

Here the prime denote derivative with respect to $||\mathbf{x} - \mathbf{y}||^2$. The first condition is a direct consequence of Mercer's condition, as before. The second condition follows by noting that the metric is simply $g_{\mu\nu} = -2\delta_{\mu\nu}K'(0)$.

1.8 Nash's Theorem: An Equivalence Class of Kernels

The following theorem holds (Nash, 1956):

Theorem 1.1

Every compact Riemannian d_L-manifold is realizable as a sub-manifold of Euclidean $(d_L/2)(3d_L + 11)$ space. Every non-compact Riemannian d_L-manifold is realizable as a sub-manifold of Euclidean $(d_L/2)(d_L + 1)(3d_L + 11)$ space.

The dimensions of the embedding spaces have since been further tightened (Greene, 1970), but Theorem 1.1 alone raises the following puzzle. Support vector machines can map to spaces whose dimension exceeds those in the Theorem. For example:

$$K(\mathbf{x}, \mathbf{y}) = e^{-||\mathbf{x}-\mathbf{y}||^2/2\sigma^2} \qquad d_F = \infty \tag{1.43}$$

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^p \qquad d_F = C_p^{p+d_L-1} \tag{1.44}$$

and in the latter case, for $(d_L = 2)$-dimensional data,

$$C_p^{p+d_L-1} > (d_L/2)(3d_L + 11) \;\; \text{for} \;\; p = 17. \tag{1.45}$$

[4] A rigorous proof of this assertion would require using suitably peaked L_2 functions, such as Gaussians, instead of Dirac delta functions, since the latter are not in L_2.

The explanation is that the map of kernels to metrics is many to one. In the left panel of Figure 1.3, the data, which lives in $\mathcal{L}$, maps via positive semidefinite kernel K (and associated mapping Φ) to $\mathbf{R}^{d_F}$ (the left hand side of the panel). This induces a metric on $\mathcal{L}$. Nash's theorem tells us that there exists a space $\mathbf{R}^d$, in which the induced metric is the same as that induced by the support vector mapping (the right hand side of the panel), and where d may be less than d_F. A concrete example is given in the right hand panel. There, one can construct a kernel implementing the identity map, which maps some subset of $\mathbf{R}^2$ to itself. In this case the metric is the (flat) Euclidean metric and the dimension of the embedding space is 2. However, a different kernel can be chosen which maps the data onto some subset of the cylinder, or to some subset of a cone (with vertex removed). In the latter two cases, the metric is still the Euclidean metric, but the minimal embedding space has dimension 3. Thus a given kernel may have an associated mapping Φ which maps to a space of higher dimension than that of the minimal dimension Euclidean space in which the input space $\mathcal{L}$, with metric arising from the kernel, can be isometrically embedded.

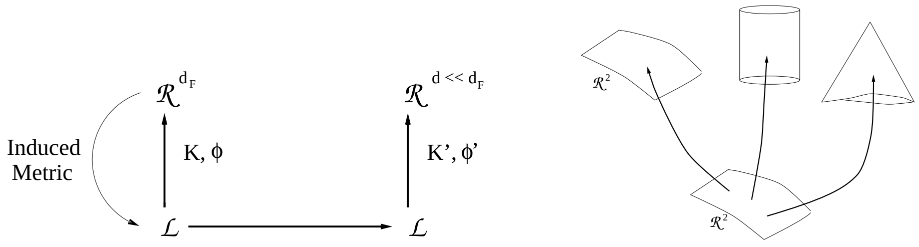

Figure 1.3 The kernels-to-metrics mapping is many-to-one.

Thus in a sense the kernel is more fundamental than the metric. An equivalence class of kernels which generate the same metric can be found by adding functions Ψ to a given kernel, where Ψ is symmetric, continuous and satisfies

$$\{\partial_{x_\mu}\partial_{x_\nu}((1/2)\Psi(\mathbf{x},\mathbf{x}) - \Psi(\mathbf{x},\mathbf{y}))\}_{\mathbf{y}=\mathbf{x}} = 0 \tag{1.46}$$

Note that, if K is a positive semidefinite kernel and $\bar{K} = K + \Psi$, for $\bar{K}$ to be also positive semidefinite, it is sufficient, but not necessary, that Ψ be positive semidefinite (sufficiency follows immediately from the linearity of Mercer's condition; as a counterexample to necessity, consider $K(\mathbf{x},\mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^p + 1$; then one can choose $\Psi = -1$, which satisfies the conditions, generates positive semidefinite $\bar{K}$, but is not itself positive semidefinite).

1.9 The Metric for Positive Definite Functions

A positive definite function $f(\mathbf{x})$ is defined as any function for which, for any n and any $c_i,\ c_j \in \mathbf{R}$ (Stewart, 1978),

$$\sum_{i,j=1}^{n} c_i c_j f(\mathbf{x}_i - \mathbf{x}_j) > 0. \tag{1.47}$$

It is known that any positive definite function is also a Mercer kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i - \mathbf{x}_j)$ (Aronszajn (1950); Stewart (1978); see also Smola et al. (1998)). Further, it is straightforward to show that any positive definite function f satisfies $f(0) \geq f(\mathbf{x})\ \forall \mathbf{x}$ (Stewart, 1978). Thus Eq. (1.42) is in concordance with known properties of positive definite functions[5].

It seems odd at first that the induced metric should take such a simple form. For example, for Gaussian Radial Basis Function kernels, $K = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$, the metric tensor becomes $g_{\mu\nu} = \delta_{\mu\nu}/\sigma^2$, which is flat. One simple consistency check is that, given two points P_1, P_2 in $\mathcal{S}$, the length of the shortest path (geodesic) in $\mathcal{S}$ which joins them must be greater than or equal to the "direct" Euclidean distance between P_1 and P_2 in $\mathcal{H}$. Let $s^2 \equiv \|\mathbf{x} - \mathbf{y}\|^2$ for two data points $\mathbf{x},\ \ \mathbf{y}\ \in \mathcal{L}$. Then the squared geodesic distance between $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ in $\mathcal{S}$ is s^2/σ^2. The Euclidean squared distance between $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ is

$$\|\Phi(x) - \Phi(y)\|^2 = 2(1 - e^{-s^2/2\sigma^2}) \tag{1.48}$$

Using the identity

$$1 - e^{-z} < z \quad \forall z \in (0, \infty] \tag{1.49}$$

with $z = s^2/2\sigma^2$, we see that the distances do at least satisfy this consistency check. As usual in infinite dimensional spaces, our intuition is unreliable: the surface $\mathcal{S}$ looks like a sphere (or a subset thereof), since $\|\Phi(\mathbf{x})\|^2 = 1\ \forall\ \mathbf{x} \in \mathcal{L}$, and the manifold $\mathcal{S}$ is a d_L-manifold, but $\mathcal{S}$ is certainly not a d_L-sphere (or a subset thereof).

1.10 A Note on Geodesics

Since support vector machines construct hyperplanes in $\mathcal{H}$, it is interesting to consider the relation between the intersection of hyperplanes with $\mathcal{S}$ and geodesics on $\mathcal{S}$. Here we offer a cautionary note: the intersection of a plane with a surface does not necessarily generate a geodesic on that surface, and the geodesics of a surface are not necessarily generated by the intersections of planes with that surface. An example shown in the Figure below illustrates these two points. The geodesics on the right circular cylinder are the generating lines, the circles, and the helices. The former two are generated by the intersection of planes with the cylinder, the latter

[5] Thanks to F. Girosi for pointing this out.

is not. On the other hand, the intersection of a plane with a right circular cylinder can generate an ellipse, which is not a geodesic.

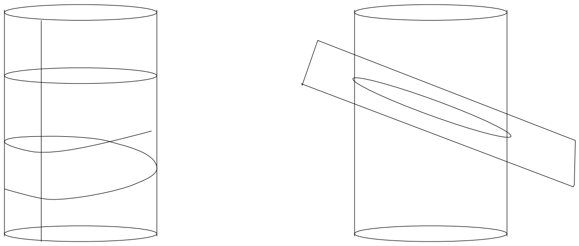

Figure 1.4 Hyperplanes may or may not generate geodesics.

1.11 Conclusions and Discussion

We have shown how, for a given positive kernel, the image of the data under the associated mapping is a Riemannian manifold (under some very general assumptions on the data and mapping), and have shown how the Riemannian metric can be computed in closed form in terms of the kernel. We have pointed out that the kernel is a more fundamental quantity than the metric, in the sense that a given metric can be generated by a large class of kernels. While the intent was to gain geometrical understanding of the kernel mapping used by the support vector and other algorithms, a useful by-product was found in the requirement that the metric be positive definite, since this gives a simple test that any proposed Mercer kernel must satisfy. A worker may wish to choose a metric on her data, and then find a Mercer kernel which generates that metric: we pointed out that there always exists such a kernel, and gave the differential equation that it must satisfy. It is hoped that the tools described here may prove useful in understanding the intrinsic geometry of the mapping associated with any positive kernel.

While here we have studied the intrinsic geometry of the manifolds to which the data is mapped, it will likely also be useful to consider the extrinsic geometry of these manifolds (i.e. the geometry arising from the particular embedding in F). For example, even though the surface in Figure 1.2 has vanishing intrinsic curvature, the extrinsic curvature is nonzero, and one would certainly expect the extrinsic geometry to play a role in the behaviour of an SVM for which F is the feature space. This will also be true for higher dimensions. However, care should be taken not to depend on properties of the embedding which do not depend on the kernel (for example, the kernel corresponding to Figure 1.2 is equally well represented by a map to 4 instead of 3 dimensions (Burges (1998)), with correspondingly different extrinsic geometry). One way to accomplish this is to ensure that any quantities one arrives at depend on the mapping Φ only through the kernel.

Part 2: Building Locally Invariant Kernels

1.12 Overview

We turn now to the question of how to incorporate known invariances of the problem into the particular kernel-based method one is using. The basic idea is as follows: given a data point $\mathbf{x}$, suppose $\mathbf{x}'$ is the result of applying a small transformation to $\mathbf{x}$, corresponding to a known symmetry of the problem (for example, translation invariance in the image classification problem). (See Smola and Schölkopf (1998) for further discussion on the kinds of symmetries one can consider). One would like one's decision function (or regression function) to give the same result on $\mathbf{x}$ and $\mathbf{x}'$, i.e., to be invariant under local (small) transformations of the data under the known symmetry. One could attempt to approach this problem by building on the metric analysis of the previous sections: that is, by choosing kernels such that $\Phi(\mathbf{x})$ and $\Phi(\mathbf{x}')$ are close, where by "close" we mean, for example, that the geodesic distance[6] joining the two transformed points is small. This could be achieved by looking for kernels whose metrics have the desired isometries.

However, the geometric analysis assumes positivity of the kernels, and many methods are not so restricted. Furthermore, even in SVM-like algorithms, the decision (or regression) function is a projected affine distance (Case III in Figure 1.1), not an intrinsic distance measured in the manifold to which the data is mapped. We will therefore instead consider the much more general problem of how to build invariances directly into the (otherwise arbitrary, and in particular, not necessarily Mercer) kernels themselves. Specifically, we consider methods that approximate some unknown function $\mathcal{G}(\mathbf{x})$ by $F(\mathbf{x})$, where $F(\mathbf{x})$ has the form:

$$F(\mathbf{x}) = \sum_q w_q K(\mathbf{p}_q, \mathbf{x}) + b, \quad w_q, b \in \mathbf{R}^1, \quad \mathbf{x} \in \mathbf{R}^n, \quad \mathbf{p}_q \in \mathbf{R}^{n'} \tag{1.50}$$

where $\mathbf{p}_q$, the weights w_q, and the threshold b are parameters that are to be determined from empirical data by a training procedure, and where the form of the kernel K is usually chosen in advance. Additive models (Hastie and Tibshirani, 1990), Radial Basis Functions (Powell, 1987; Girosi et al., 1995; Bishop, 1995), and Support Vector Machines (Cortes and Vapnik, 1995; Vapnik, 1995) are examples of such methods. Pattern recognition, regression estimation, density estimation, and operator inversion are examples of problems tackled with these approaches (Vapnik, 1995). Thus for example in the density estimation case, $\mathbf{x}$ would be a point (in a vector space) at which the probability density is required, and $F(\mathbf{x})$ would be the approximation to that density; for the classification case, $\mathbf{x}$ would be a test pattern

[6] Of course, we mean the smallest geodesic distance here (e.g. two points on a 2-sphere are connected by two geodesics, which together make up the great circle on which the points lie).

to be classified, and $\mathrm{sgn}(F(\mathbf{x}))$ would give the corresponding label.

One can also view the problem as that of model selection: given a task, how can one find a family of kernels that is best suited to that task? While results are known for some particular kernel based methods (for example, for SVMs, model selection based on VC bounds has been investigated by Schölkopf et al. (1995), and more recently, Schölkopf et al. (1998a) describe some methods for incorporating prior knowledge in SVMs), it is of interest to ask how one might restrict the class of models, using the prior knowledge of a known symmetry of the problem, for arbitrary kernel-based methods.

Section 1.13 describes the general approach. In Section 1.14, we take the example of vertical translation invariance in images to perform a detailed analysis for a particular case.

1.13 Incorporating Local Invariances

Given two test points $\mathbf{x}$ and $\mathbf{y}$, equation (1.50) defines a distance function

$$\rho(\mathbf{x},\mathbf{y}) = \sum_q w_q(K(\mathbf{p}_q,\mathbf{x}) - K(\mathbf{p}_q,\mathbf{y})). \tag{1.51}$$

Note that $\rho(\mathbf{x},\mathbf{y})$ is an affine distance, in that it can take positive or negative values. For example, ρ can be thought of as counting contours between patterns in the pattern recognition case.

We would like to choose a class of kernels so that $\rho(\mathbf{x},\mathbf{y})$ is close to zero if $\mathbf{y}$ is a transform of $\mathbf{x}$ along some symmetry direction. If $\mathbf{y} = \mathbf{x} + d\mathbf{x}$, then

$$d\rho = \sum_{q,i} w_q dx^i \partial_i K(\mathbf{p}_q,\mathbf{x}) \tag{1.52}$$

Local Invariance

where we define $\partial_i \equiv \partial/\partial x_i$. Requiring that this be zero for all w_q gives:

$$\sum_i dx^i \partial_i K(\mathbf{p}_q,\mathbf{x}) = 0. \tag{1.53}$$

Note that for a particular problem, for which the w_q are known to satisfy certain constraints, equation (1.53) may be more restrictive than is necessary to ensure that $d\rho = 0$, but in this work we will make no assumptions about the w_q.

We write a general one-parameter transformation as:

$$x'_i = x_i + \alpha f_i(\mathbf{x}), \;\; \alpha \in R^1 \tag{1.54}$$

for which, in the limit as $\alpha \to 0$, Eq. (1.53) takes the form:

$$\sum_i f_i(\mathbf{x})\partial_i K(\mathbf{p}_q,\mathbf{x}) \equiv \mathcal{O}K(\mathbf{p}_q,\mathbf{x}) = 0 \tag{1.55}$$

which defines the operator $\mathcal{O}$. Henceforth we will not explicitly write the parameter vector $\mathbf{p}_q$ in K.

1.13.1 Operator Invariance for the Linear Case

We prove two simple propositions for the case in which the transformation (1.54) is both linear and invertible.

Proposition 1.3
For linear, invertible transformations (1.54), the operator $\mathcal{O}$ is itself invariant under the transformation.

Proof Let $\mathbf{U}$ denote the unit matrix, T denote transpose, and ∂ denote the vector with components $\partial_i \equiv \partial/\partial x_i$. Denote the transformation by

$$\mathbf{x}' = (\mathbf{U} + \alpha\mathbf{M})\mathbf{x}. \tag{1.56}$$

Then the operator $\mathcal{O}$ in Eq. (1.55) is given by:

$$\mathcal{O} = \mathbf{x}^T\mathbf{M}^T\partial \tag{1.57}$$

and

$$\mathcal{O}' = \mathbf{x}'^T\mathbf{M}^T\partial' = \sum_{i,j,k} M_{ij}(U+\alpha M)_{jk}x_k(U+\alpha M)^{-1}_{mi}\partial_m = \mathbf{x}^T\mathbf{M}^T\partial = \mathcal{O} \tag{1.58}$$

■

Proposition 1.4
For linear, invertible transformations (1.54), denoting the argument of $K()$ by its components x_i, if $K(x_i)$ satisfies (1.55), then so does $K(x_i + \alpha f_i(\mathbf{x}))$, for *finite* α.

Proof $\mathcal{O}K(x_i) = 0 = \mathcal{O}'K(x_i + \alpha f_i(\mathbf{x})) = \mathcal{O}K(x_i + \alpha f_i(\mathbf{x})),$ (1.59)
since $\mathcal{O}$ is an invariant by the above theorem. ■

1.13.2 Multiple Symmetries

For a set of M symmetries, there will be M simultaneous equations of the form (1.55):

$$\mathcal{O}_m u(x_0, \ldots, x_{n-1}) = 0, \quad m = 1, \ldots, M \tag{1.60}$$

where $\{\mathcal{O}_m\}$ is a set of linear differential operators, each of which takes the general form

$$\mathcal{O}_m = \sum_i f_{mi}(\mathbf{x})\partial_i. \tag{1.61}$$

Thus in order to find the set of kernels which are simultaneously invariant under multiple symmetries, we need to find all the non-trivial integrals of (1.60). Following Chester (1971), we have:

Definition 1.1
Complete Operators

A system of operators $\{\mathcal{O}_i\}$ is called *complete* if all commutators take the form

$$[\mathcal{O}_i, \mathcal{O}_j] = \sum_k c_{ijk}\mathcal{O}_k, \quad c_{ijk} \in \mathbf{R}^1 \tag{1.62}$$

Note that, from their definition, the c_{ijk} are skew symmetric in i, j and satisfy the Jacobi identity, so they are in fact the structure constants of a Lie algebra (Olver, 1986).

Definition 1.2

A system of equations (1.60) is called *complete* if the corresponding operators form a complete set (Chester, 1971).

Complete Equations

Theorem 1.2

Any complete system of $r < n$ independent equations, of the form (1.60), has exactly $n - r$ independent integrals (Chester, 1971).

Thus, non-trivial integrals u will only exist if the number of operators r in the complete set is less than n, and if so, there will be $n - r$ of them. In the latter case, the general solution of the system (1.60) will have the form

$$F(u_0(x_0, \ldots, x_{n-1}), \ldots, u_{n-r-1}(x_0, \ldots, x_{n-1})) \tag{1.63}$$

where F is any C^1 function. Note that in order to find the complete set of operators one simply generates new operators by computing all the commutators in (1.62). If the number of independent operators thereby found is less than n, one has a complete set; otherwise there exists no non-trivial solution.

Imposing constraints on the kernel functions may thus be viewed as a form of capacity control, where the number of degrees of freedom in the problem is explicitly reduced by the dimension of the Lie algebra in Eq. (1.62).

1.13.3 Building Locally Invariant Kernels

Given the general solution to the system (1.60), we can now easily construct locally invariant kernels, since although the functions F must be differentiable, they are otherwise arbitrary. Henceforth, we will use $\mathcal{I}$ ("input") to label the space $\mathbf{R}^n$ in which the data are elements, and $\mathcal{P}$ ("preprocessed") to label the space $\mathbf{R}^{n-r}$, in which the independent integrals $\mathbf{u}$ may be viewed as coordinates. Thus one can view the mapping from $\mathcal{I}$ to $\mathcal{P}$ as a preprocessing step, after which a known family of kernels may be applied. Note that the kernels may have further constraints placed on them by the particular method being used (for example, their dependence on the parameter set $\mathbf{p}_q$), but that this is easily dealt with, since any of the original set of kernels can be used (but on the preprocessed data).

For example, for support vectors machines, the kernels take the form $K(\mathbf{s}_q, \mathbf{x})$, $\mathbf{s}_q$, $\mathbf{x} \in \mathbf{R}^n$, where the $\mathbf{s}_q$ are the support vectors. The kernels must be symmetric, $K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x}) \;\; \forall \;\; \mathbf{x}, \mathbf{y} \in \mathbf{R}^n$, and continuous, and they must satisfy Mercer's constraint, Eq. (1.2). Now suppose that some number of symmetries have been imposed, resulting in a complete set of size $r < n$, so that the number of independent integrals (and hence, the dimension of $\mathcal{P}$) is $n-r$. Thus, the solutions have the general form (1.63) above. We can then simply choose F to have the form of a kernel function which is known to satisfy the constraints, but which takes $n-r$ arguments instead of n. For example, we might take degree p polynomial kernels, for which the F will have the form

$$F(\mathbf{u},\mathbf{v}) = (\mathbf{u}\cdot\mathbf{v}+1)^p, \quad \mathbf{u},\mathbf{v} \in \mathbf{R}^{n-r} \tag{1.64}$$

Since such kernels are symmetric, continuous and satisfy Mercer's condition, all the desired constraints are satisfied. What was a polynomial support vector machine has become a polynomial support vector machine acting on preprocessed data. However, the functional form that the overall kernels take, when considered as functions of the input data $\mathbf{x}$, will no longer in general be polynomial. However it may be necessary to use kernels other than those which give good performance on data in $\mathcal{I}$, as we shall see below.

1.14 A Detailed Example: Vertical Translation Invariance

Let us use the example of vertical translation invariance for the pattern recognition problem in order to explore in detail where the ideas of the previous Section lead. We consider an image of n rows and one column, and we impose cyclic boundary conditions. We do this because solving this problem amounts to solving the general n_1 by n_2 problem: an n_1 by n_2 image can be converted to an $n_1 n_2$ by 1 image by pasting the top of each column of pixels to the bottom of the previous column. Finally, we consider transformations corresponding to shifts of up to one pixel. (To make the cyclic boundary conditions a realistic assumption, a border of one blank pixel could be added to the top and bottom of the original image.) In this case the transformations (1.54) take the form:

$$x_i{}' = (1-\alpha)x_i + \alpha x_{(i+1)}, \quad i = 0,\ldots,N-1, \quad \alpha \in [0,1] \tag{1.65}$$

Note that here and below we adopt the convention that indices i, j, k are to be taken modulo n. Eq. (1.55) becomes:

$$\sum_{i=0}^{n-1} \{(x_{i+1} - x_i)\partial_i\} u(\mathbf{x}) = 0 \tag{1.66}$$

1.14.1 The Relation to Group Action

We start by exploring the relation of the transformations (1.65) to a group action. Since pixels are an approximation to a continuous function which can be considered as a representation of the group of translations, let us first make the connection between pixels and the continuous case. Let $I(z)$ represent the original image field for which the n-pixel column is an approximation. Thus

$$x_i = I(i),\ \ i = 0, \ldots, n-1 \tag{1.67}$$

Translating by a distance α means replacing $I(z)$ by $I'(z) = I(z-\alpha)$. The new pixel values become

$$x'_i = I'(i) = I(i-\alpha) \approx I(i) - \alpha(\partial_z I)(i) \tag{1.68}$$

Approximating $(\partial_z I)(i)$ by $I(i) - I(i-1)$ then gives equation (1.65).

However, the binning of the data into a vector has the consequence that Eq. (1.65), for finite α, is not a group action, although it does constitute the desired transformation. We illustrate this point with a simple specific case, namely vertical translation invariance for a column of three pixels, whose values we label by x, y, z. Then (1.65) for x becomes:

$$x' = g_\alpha x \equiv x(1-\alpha) + \alpha y \tag{1.69}$$

where g_α is the operator instantiating the transformation. Then

$$(g_\beta \circ g_\alpha)x = x(1-\alpha)(1-\beta) + y(\alpha + \beta - 2\alpha\beta) + z\alpha\beta \tag{1.70}$$

Infinitesimal Group Action

so there exists no γ such that $g_\gamma = g_\beta \circ g_\alpha$. However, to first order in α, β, the above transformations do form a group, with $g_\alpha^{-1} = g_{-\alpha}$. Thus, the action may be viewed as of a group only for infinitesimal values of the parameters (Olver, 1986). Despite this fact, the transformation (1.69) does constitute a translation of the binned data for finite values of α (in fact for any $\alpha \in [0,1]$).

Generators

But if the action is a group action for infinitesimal values of the parameters, then the corresponding differential operators are necessarily generators for a one-parameter Lie group. However the representation of that group for finite values of α does not coincide with (1.65). One can find what the representation is by exponentiating the generator corresponding to Eq. (1.65) acting on a particular point, for example:

$$\begin{aligned} e^{\{\alpha((y-x)\partial_x + (z-y)\partial_y + (x-z)\partial_z)\}}x &= x + (y-x)h_1 + (x-2y+z)h_2 + \\ &+(y-z)h_3 + (x+y-2z)h_4 + (x-z)h_5 + (-2x+y+z)h_6 \end{aligned} \tag{1.71}$$

where the h_i are functions of α alone. This only corresponds to the transformation (1.69) to first order in α. To summarize, the transformation (1.69) coincides with that of a Lie group only for infinitesimal values of the parameters; for finite values, it is no longer a group, but it is still the desired transformation.

1.14.2 A Simple Example: 4 Pixels

Let us find the complete solution for the next simplest case: an image which consists of just 4 pixels. Eq. (1.66) becomes:

$$\{(x_1 - x_0)\partial_{x_0} + (x_2 - x_1)\partial_{x_1} + (x_3 - x_2)\partial_{x_2} + (x_0 - x_3)\partial_{x_3}\}u(\mathbf{x}) = 0. \tag{1.72}$$

The general solution to this is:

$$f(x_0, x_1, x_2, x_3) = F(u_0, u_1, u_2) \tag{1.73}$$

where

$$u_0 = x_0 + x_1 + x_2 + x_3 \tag{1.74}$$

$$u_1 = \ln\left(\frac{x_0 - x_1 + x_2 - x_3}{(x_0 - x_2)^2 + (x_3 - x_1)^2}\right) \tag{1.75}$$

$$u_2 = \arctan\left(\frac{x_3 - x_1}{x_0 - x_2}\right) + \\ + \frac{1}{2}\ln((x_0 - x_2)^2 + (x_3 - x_1)^2) \tag{1.76}$$

where F is any C^1 function. Thus to use this solution in a kernel based method, one would replace F by one's choice of kernel, but it would be a function of the three variables $u_0,\ u_1,\ u_2$ instead of the four x_i.

This solution has two properties to which we wish to draw attention: First, u_0, and only u_0, is "globally" invariant (invariant for any of the allowed values of α in Eq. (1.65); u_0 corresponds to the "total ink" in the image); second, all three independent integrals have a property which we call "additive invariance".

1.14.2.1 Additive Invariance

Recalling that the transformed variables are denoted with a prime (Eq. (1.65)), we define an "additive invariant" integral to be one that has the property:

$$u_j(\mathbf{x}') = u_j(\mathbf{x}) + f_j(\alpha), \quad j = 0, \ldots, 2 \tag{1.77}$$

for some functions f_j. Clearly, by construction, $f_j(\alpha)$ is $O(\alpha^2)$.

Additive invariance reduces the number of degrees of freedom in the problem in the following sense: consider two points in input space $\mathcal{I}$ which are related by a vertical translation. They will map into two points in $\mathcal{P}$, whose difference is independent of the original data, and which depends only on the transformation parameter α. Thus to learn that two images are translates of each other, the learning machine, acting on $\mathcal{P}$, has only to learn a vector valued function of one parameter $(f_j(\alpha))$, *where that parameter is independent of the original data* (i.e. independent of the pixel values of the original image). Note that the original data does not have this property: for pixel data, the difference between two translated images depends on the original image. If we were able to find solutions with $f_j(\alpha) = 0$, we would have a global invariance. However, we shall prove below that, for the symmetry and boundary conditions considered, there is only one globally invariant solution, for arbitrary dimension n; thus, in the absence of a family of globally invariant solutions, additive invariance appears to be a desirable property for the solutions to have.

Note that the solution of a PDE has a large degree of arbitrariness in the independent integrals found. For the above example, we could equally well have taken u_j^2 instead of the u_j as the integrals. The former do not have the additive invariance property. Thus for a general problem, we can only hope to find particular additive invariant integrals, rather than prove that all integrals will be additive invariant.

Note also that additive invariance only strictly holds if the two images $\mathbf{x}$ and $\mathbf{x}'$ being compared have values of the arctan in (1.76) which do not lie on opposite sides of a cut line. If this does not hold, then there will be an additional 2π in $u_2(\mathbf{x}) - u_2(\mathbf{x}')$. While this extra term does not depend continuously on the data, it is nevertheless a dependence.

Finally, we point out that u_0 and u_1 above are also solutions of the corresponding differential equations for vertical translation invariance in the opposite direction. Since the two operators commute, by Theorem 1.2 we know that there will be a total of two independent integrals, so we know that u_0 and u_1 constitute the full solution, and this solution is clearly additive invariant. This example also illustrates a property of invariances in binned data which differs from that for continuous data: for continuous data, vertical translations compose a one-parameter, Abelian group. For binned data, translating "down" gives different generators from translating "up", which will give different generators from translating up by 2 pixels instead of 1, and so forth. Each of these imposed symmetries will reduce the dimensionality of $\mathcal{P}$ by one.

The example given in this Section raises the following questions for the case of arbitrarily sized images: first, assuming that the general solution can be found, will we still only be able to find one *globally* invariant independent integral (analogous to u_0 above)? If so, additive invariant solutions will perhaps be useful. Second, can one construct solutions so that all independent integrals are additively invariant? The answers to both of these questions is yes, as we shall now show.

1.14.3 The n-pixel case

As mentioned above, in order to answer the above questions for an arbitrarily sized image, we only need consider the case of an n-pixel stack, and we must then solve Eq. (1.66). We start, however, by answering the first question above.

1.14.3.1 A No-Go Theorem

We have the following

Theorem 1.3
Any solution of (1.66), which is also invariant under the transformation (1.65) for any $\alpha \in [0, 1]$, has the form $F(\sum_i x_i)$, where $F \in C^1$.

The proof is given in the Appendix. This theorem demonstrates that we cannot hope to find globally invariant solutions of Eq. (1.66), other than $F(\sum_i x_i)$. Thus, we will need to search for "second best" solutions, such as additive invariant ones.

1.14.3.2 The General Solution

We now derive the general solution for the n-pixel case. Eq. (1.66) may be viewed as the dot product of the gradient of an unknown function u with a vector field in n dimensions, and to find the general solution one must solve the set of ODE's which describe the characteristic surfaces, which are themselves parameterized by some $t \in \mathbf{R}^1$ (Zachmanoglou and Thoe, 1986):

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} \tag{1.78}$$

where

$$A_{ij} = -\delta_{ij} + \delta_{i,j+1}. \tag{1.79}$$

$\mathbf{A}$ has determinant zero and eigenvalues λ_k given by

$$1 + \lambda_k = e^{2\pi ik/n}, \quad k = 0, \cdots, n-1 \tag{1.80}$$

Here and for the remainder, we reserve the symbol i for $\sqrt{-1}$. By inspection we can construct the eigenvectors $\mathbf{z}$ of the matrix $\mathbf{A}$

$$z_{k,j} = e^{2\pi ijk/n}, \quad j, k = 0, \cdots, n-1 \tag{1.81}$$

where the first index k labels the eigenvector, and the second j its components. Let $\mathbf{S}$ be the matrix whose columns are the eigenvectors $\mathbf{z}$. Then $\mathbf{S}$ diagonalizes $\mathbf{A}$:

$$\mathbf{S}^{-1}\mathbf{A}\mathbf{S} = \mathrm{diag}(\lambda_i) \tag{1.82}$$

One can confirm by multiplication that the inverse of $\mathbf{S}$ is given by $(1/n)\mathbf{S}^\dagger$ (where $\dagger$ denotes hermitian conjugate). Thus introducing $\mathbf{y} \equiv \mathbf{S}^{-1}\mathbf{x}$, the solutions of equations (1.78) are given by

$$y_0 = c_0 \tag{1.83}$$

$$t = \frac{1}{e^{2\pi ik/n} - 1}\ln(\frac{y_k}{c_k}), \quad k = 1, \cdots, n-1 \tag{1.84}$$

where the c_k are constants of integration. One can then easily show that expressions (1.83) and (1.84) constitute only $n-1$ independent equations.

We can now write the explicit solution to (1.66), which we do for n even (the solution for n odd is similar). Again, let F be any C^1 function. The general solution may be written

$$f(x_0, \cdots, x_{n-1}) = F(u_0, \cdots, u_{n-2}) \tag{1.85}$$

where

$$u_0 = \sum_i x_i \tag{1.86}$$

$$u_{2k-1} = (1/\phi_t)\arctan(\phi_s/\phi_c) + (1/2)\ln(\phi_c^2 + \phi_s^2) \tag{1.87}$$

$$u_{2k} = (1/2)\ln(\phi_c^2 + \phi_s^2) - \phi_t \arctan(\phi_s/\phi_c) - \ln T \tag{1.88}$$

and where $k = 1, \cdots, (n/2) - 1$, and we have introduced

$$\phi_s(n, k, \mathbf{x}) \equiv \sum_{j=0}^{n-1} \sin(2\pi kj/n) x_j \tag{1.89}$$

$$\phi_c(n, k, x) \equiv \sum_{j=0}^{n-1} \cos(2\pi kj/n) x_j \tag{1.90}$$

$$\phi_t(n, k) \equiv \frac{\sin(2\pi k/n)}{\cos(2\pi k/n) - 1} \tag{1.91}$$

and

$$T \equiv \sum_{j=0}^{n-1} (-1)^j x_j = \phi_c(n, n/2, \mathbf{x}). \tag{1.92}$$

1.14.3.3 Additive Invariance

We now show that all the independent integrals in the above solution have the additive invariance property, up to factors of 2π which result from the cut line of the arctan function. Clearly u_0 is additive invariant (in fact it is invariant). The u_k, $k > 0$ in (1.85) were all obtained by taking real and imaginary parts of linear combinations of equations (1.84), i.e. of

$$t = \frac{1}{e^{2\pi k/n} - 1} \ln\{(1/nc_k) \sum_j e^{-2\pi ikj/n} x_j\}, \quad k > 0 \tag{1.93}$$

Transforming the $\mathbf{x}$ according to (1.65) gives the transform of t:

$$t_\alpha = \left(\frac{1}{e^{2\pi k/n} - 1}\right) \ln(1 - \alpha + \alpha e^{2\pi ik/n}) + t, \quad k > 0 \tag{1.94}$$

Thus taking linear combinations of these equations will always give equations which separate into the sum of an α-dependent part and an $\mathbf{x}$-dependent part. Hence all solutions in (1.87), (1.88) (and (1.86)) are additive invariant.

1.15 The Method of Central Moments

It is interesting to compare the method described above, for the case of translation invariance, with the method of central moments, in which translation invariant features are constructed by taking moments with respect to coordinates which are relative to the center of mass of the image. From these moments, rotation and scale invariant moments can also be constructed. For example, for continuous data, one can construct the translation invariant moments (Schalkoff, 1989):

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\hat{x})^p(y-\hat{y})^q I(x,y)dxdy, \tag{1.95}$$

where x, y are the coordinates of points in the image, $\hat{x}$, $\hat{y}$ are the coordinates of the center of mass, and $I(x,y)$ is the corresponding image intensity. To apply this to pixel data one must replace (1.95) by an approximate sum, which results in features which are only approximately translation invariant (although as the number of pixels increases, the approximation improves). However, for binary data, $I \in \{0,1\}$, it is possible to construct moments from the pixel data which are exactly invariant:

$$\mu_{pq} = \sum_{j,k\neq 0}(j-\hat{j})^p(k-\hat{k})^q I(j,k) \tag{1.96}$$

where the sum is over only non-zero values of i and j.

How does the existence of these invariants relate to the above theorem on the existence of only one globally invariant integral for vertical translation invariance with cyclic boundary conditions? The μ_{pq} in Eq. (1.95) are globally invariant, but there the data must be not binned, and are continuous; the μ_{pq} in Eq. (1.96) are also globally invariant, but there the data must be binned, and discrete. Thus our theorem (for binned, continuous data) addresses the case lying between these two extremes. (However, it is an open question whether Theorem 1.3 holds also for other choices of boundary conditions).

The method of central moments, compared to the approach described above, has the advantage that the features are likely to be less sensitive to motions in the symmetry directions, especially when the number of pixels is very large. However, it is not clear how many moments, and which moments, are needed to give good generalization performance for a given problem. Furthermore, the problem of which kernels to use on the preprocessed data remains, and is critical, as we see below.

1.16 Discussion

We have explored a very general approach to incorporating known symmetries of the problem in kernel-based methods, by requiring that the kernels themselves be invariant (at least locally) under that set of transformations under which one expects the function (Eq. (1.50)) to be invariant. We have used the case of translation invariance of binned data with cyclic boundary conditions to explore these ideas in detail. For that case, we showed that one cannot hope to construct globally invariant kernels which depend on anything other than the sum of the pixel data. We also showed that the mapping can be chosen to have the property that the difference of two mapped images, which are translates of each other, is independent of the original data. Since the class of globally invariant functions is too small to be of much use, this "additive invariance" would seem to be an attractive property, although to benefit one would have to choose kernels (acting on $\mathcal{P}$), which take advantage of it.

However, experiments done on NIST 28x28 digit data, using polynomial and RBF support vector machines, and also a kernel version of Kth nearest neighbour, acting on data preprocessed according to Eqs. (1.86), (1.87), and (1.88), showed no improvement in generalization performance: in fact, most experiments gave considerably worse results. Since all the original pixel information (minus a degree of freedom due to translation invariance) is still in the preprocessed data, we conclude that these particular kernels, which work well for support vector machines acting on pixel data, are not well suited to the preprocessed data. Perhaps this should not be surprising: after all, a polynomial kernel computes correlations between pixels directly; it is not clear what a polynomial kernel acting on $\mathcal{P}$ is doing. Thus further research is needed regarding how to find suitable kernels for the preprocessed data (in particular, ones which take advantage of additive invariance). But whatever kernels are used, we know that the resulting system will have the desired local invariance.

The theory described above still leaves a great deal of freedom in the form the solutions take. For example in the case studied, the independent integrals, Eqs. (1.86) - (1.88), form the basis for a general solution, but this basis is itself far from unique. It is straightforward to find alternative sets of independent integrals. In one such set, the discrete fourier transform (DFT) character of the transformation can be made explicit, in the sense that if the highest frequency component of the input data is normalized so that $T = 1$ in (1.92), the transformation becomes an exact DFT. Since the DFT has an inverse, this example makes explicit the fact that the effect of preprocessing is to remove just one degree of freedom. The large class of possible solutions raises a similar question to the one above: if one chooses a different set of independent integrals for a given invariance, must one also choose different subsequent kernels to achieve the same generalization performance?

Finally, we have only required that the first derivative of the kernel vanish along a symmetry direction. This is a weak condition: a kernel that satisfies this may still vary significantly when data is transformed, even by small amounts, along that symmetry direction. One could proceed by requiring that higher derivatives also vanish. However, one would still require that the first derivative vanish, so the analysis carried out above would still pertain.

1.17 Appendix

Proof of Theorem 1.3:

We introduce the notation "prm" to denote cyclic permutation of the indices. In the following, indices i, j, k are always to be taken modulo n. By definition, an invariant solution must satisfy $\partial_\alpha u(\mathbf{x}') = 0$, where $\mathbf{x}'$ is the linear function of α defined in (1.65). However,

$$0 = \partial_\alpha u(\mathbf{x}') = (x_1 - x_0)\partial_0' u(\mathbf{x}') + \text{prm} \tag{1.97}$$

Here we have introduced the notation $\partial_\alpha \equiv \partial/\partial\alpha$ and $\partial_1' \equiv \partial/\partial x_1'$, etc. We can generate a set of PDEs that u must satisfy by expressing the x_i in terms of the x_i' in Eq. (1.97). Note first that the transformations (1.65) can be written

$$\mathbf{x}' = \mathbf{M}\mathbf{x} \tag{1.98}$$

where

$$M_{ij} \equiv \delta_{i,j}(1-\alpha) + \delta_{i,j-1}\alpha, \quad i,j = 0,\cdots,n-1 \tag{1.99}$$

Note also that

$$\det(\mathbf{M}) \equiv S = (1-\alpha)^n - (-1)^n\alpha^n. \tag{1.100}$$

One can verify directly, by matrix multiplication, that

$$(M^{-1})_{ij} = (1/S)(\alpha-1)^{i-j-1}\alpha^{j-i}(-1)^{n-1} \tag{1.101}$$

(recall that subexpressions containing i, j, k are to be taken modulo n, so that the exponents in (1.101) only take values between $0,\cdots,n-1$). Thus Eq. (1.98) gives

$$(-1)^{n-1}S(x_1 - x_0) = (\alpha^{n-1} - (\alpha-1)^{n-1})x_0' - \sum_{j=1}^{n-1}(\alpha-1)^{n-j-1}\alpha^{j-1}x_j' \tag{1.102}$$

By using the fact that both $\mathbf{M}$ and $\mathbf{M}^{-1}$ are cyclic matrices, it is straightforward to show that the expression for $x_{i+1} - x_i$ can be obtained from that for $x_1 - x_0$ in Eq. (1.102) by replacing x_k' by x_{k+i}' on the right hand side (and leaving other terms unchanged). We need the following

Lemma 1.1
Extracting the coefficients of powers of α on the right hand side of (1.97) gives the family of PDEs

$$\begin{aligned} ((x_1' - x_0')\partial_0' + \text{prm})u(\mathbf{x}') &= 0 \\ ((x_2' - x_0')\partial_0' + \text{prm})u(\mathbf{x}') &= 0 \\ &\cdots \\ ((x_{N-1}' - x_0')\partial_0' + \text{prm})u(\mathbf{x}') &= 0 \end{aligned} \tag{1.103}$$

Proof The proof is by induction. First we note by inspection that the coefficient of α^0 on the right hand side of (1.102) is $(-1)^n x_0' - (-1)^{n-2}x_1'$. Thus substituting (1.102) in (1.97) and taking the coefficient of $O(1)$ gives

$$\{(x_1' - x_0')\partial_0' + \text{prm}\}u(\mathbf{x}') = 0. \tag{1.104}$$

Equation (1.104) can then be substituted in (1.97) to eliminate terms of order α^0. By repeating the process with coefficients of $O(\alpha)$, one arrives at an equation which one can combine with (1.104) (to eliminate x_1' in the first term) to get

$$\{(x_2' - x_0')\partial_0' + \text{prm}\}u(\mathbf{x}') = 0. \tag{1.105}$$

Now assuming that this works up to α^p (giving the first $p+1$ equations in (1.103)), we must show that it works for α^{p+1} (giving the next equation in (1.103)). The coefficient of ∂_0' in (1.97) is the right hand side of Eq. (1.102) (we have divided everything by the overall factor $(-1)^{n-1}S$). Using the first $p+1$ equations in (1.103), we can effectively replace x_1' by x_0', x_2' by x_0' etc., in the first term on the right hand side of Eq. (1.97). Doing this means that the coefficient of ∂_0' in (1.97) becomes

$$\begin{aligned}&\{\alpha^{n-1} - (\alpha-1)^{n-1} - (\alpha-1)^{n-2} - \alpha(\alpha-1)^{n-3} - \\ &\cdots - \alpha^p(\alpha-1)^{n-(p+2)}\}x_0' - \sum_{j=p+2}^{n-1}(\alpha-1)^{n-j-1}\alpha^{j-1}x_j'\end{aligned} \tag{1.106}$$

Using the identity

$$\sum_{i=1}^{p+1} C_i^{n-3-p+i} = C_{p+1}^{n-1} - 1 \tag{1.107}$$

we find that the coefficient of α^{p+1} in (1.106) is $(-1)^{N-p-2}(-x_0' + x_{p+2}')$. Hence we have generated the $p+2$'th equation in Eq. (1.103). This completes the proof of the Lemma. ■

Now note that (1.103) are independent equations, since the matrix of coefficients is of rank $n-1$ for some choices of the x_i. Finally, it is straightforward to check that all the operators appearing in (1.103) commute, so this set of $n-1$ PDEs forms a complete set. Since it is a complete set of $n-1$ PDEs in n dimensions, by Theorem 1.2 it has only one integral solution. By substitution it is easily checked that $u(\mathbf{x}) = \sum_i x_i$ is a solution; thus the general solution of (1.97) must take the form $F(\sum_i x_i)$, where $F \in C^1$. This completes the proof of the theorem.

1.18 Acknowledgements

I wish to thank P. Knirsch, C.R. Nohl, B. Schölkopf, A. Smola, C. Stenard and V. Vapnik for useful discussions. This work was funded in part by ARPA contract N00014-94-C-0186.

References

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337 – 404, 1950.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

William M. Boothby. *An introduction to differentiable manifolds and Riemannian geometry.* Academic Press, 2nd edition, 1986.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector learning machines. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 375–381, Cambridge, MA, 1997. MIT Press.

C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):1–47, 1998.

Clive R. Chester. *Techniques in Partial Differential Equations.* McGraw Hill, 1971.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273 – 297, 1995.

R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers, Inc, New York, 1953.

K. Dodson and T. Poston. *Tensor Geometry.* Springer-Verlag, 2nd edition, 1991.

E.T.Copson. *Metric Spaces.* Cambridge University Press, 1968.

F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

R.E. Greene. *Isometric Embeddings of Riemannian and Pseudo-Riemannian Manifolds.* American Mathematical Society, 1970.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability.* Chapman & Hall, London, 1990.

A.N. Kolmogorov and S.V.Fomin. *Introductory Real Analysis.* Prentice-Hall, Inc., 1970.

J. Nash. The embedding problem for riemannian manifolds. *Annals of Mathematics*, 63:20 – 63, 1956.

P. J. Olver. *Applications of Lie Groups to Differential Equations.* Springer-Verlag, 1986.

M.J.D. Powell. Radial basis functions for multivariable interpolation: A review. In *Algorithms for Approximation, J.C. Mason and M.G. Cox (Eds.)*, pages 143–167. Oxford Clarendon Press, 1987.

Robert J. Schalkoff. *Digital Image Processing and Computer Vision.* John Wiley and Sons, Inc., 1989.

B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings, First International Conference on Knowledge Discovery & Data Mining.* AAAI Press, Menlo Park, CA, 1995.

B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, Cambridge, MA, 1998a. MIT Press.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299 – 1319, 1998b.

B. Schölkopf, A. Smola, K.-R. Müller, C. Burges, and V. Vapnik. Support vector methods in learning and feature extraction. In T. Downs, M. Frean, and M. Gallagher, editors, *Proceedings of the Ninth Australian Conference on Neural Networks*, pages 72 – 78, Brisbane, Australia, 1998c. University of Queensland.

A. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 1998. In press.

A. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 1998. In press.

James Stewart. Positive definite funcions and generalizations, an historical survey. *Rocky Mountain Journal of Mathematics*, 6(3):409–434, 1978.

V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.

V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 281–287, Cambridge, MA, 1997. MIT Press.

E.C. Zachmanoglou and Dale W. Thoe. *Introduction to Partial Differential Equations with Applications.* Dover, Mineola, N.Y., 1986.