Stichting Onderzoek Wereldvoedselvoorziening van de Vrije Universiteit

Centre for World Food Studies

Kernel learning for poverty mapping: an introduction

by

M.A. Keyzer

Staff Working Paper **WP – 02 – 08** **July 2002**

Contents

Abstract

Kernel learning offers tools for semiparametric regression and classification that have made their proofs in the field of pattern recognition, especially in optical reading, voice recognition, and genomics (Schölkopf and Smola, 2002). In this paper, we describe its possible contribution to the construction of poverty maps, especially its potential to improve the flexibility of the functional forms in regression and shows how to use census information for estimating these. We have implemented the kernel learning algorithms in GAMS and found to be numerically effective, as they essentially rely on convex quadratic programming.

Section 1
Introduction

Poverty maps are constructed via integration, by geographical district, of a function $f(x)$ estimated on the basis of a given data set, a survey, with data on the vector of independent variables x and the scalar dependent variable y associated to f, over a wider data set, a census, that contains data on x only, so as to obtain estimates of the means of f for each district (Hentschel et al. 2000, Tarozzi, 2001). Kernel learning offers a semi-parametric procedure to identify an unknown regression function $f(x)$ but it has so far not been used in such a poverty mapping exercise.

This paper highlights its potential to improve the flexibility of the functional forms applied in the regression of $f(x)$, and shows how to use census information for estimating these. It largely draws on the recent book by Schölkopf and Smola (2002, S&S, for short), that provides a good overview of the state of the art in this field.

Regression analysis postulates that a given set of observations can to a great extent be explained by a function $f(x)$, leaving small residuals. At similar x-values, f observations should lie close to $f(x)$. This raises two questions. The first is how to specify a measure of dissimilarity between x-values. For example, in one context the geographical distance between two points may be the relevant criterion, whereas in another the distance along a practicable road would be the appropriate measure. The second question is how to formulate a class of functions from which $f(x)$ is to be selected, while relying on well established function expansions rather than on trial-and-error search for the best functional form. Remarkably, kernel learning so far has hardly penetrated applied econometrics, even though it is particularly suited to both tasks.[1]

First, kernel learning has been successful in pattern recognition (for script, voice, or images and genomics) in situations where the number of observations is modest relative to the number of variables and with quantitative as well as qualitative observations. Its strength essentially derives from the kernel's ability to attribute appropriate weights to the observations available based on a priori information on distance measures.

Second, kernel learning offers a unified conceptual framework that includes support vector regression as well as clustering and classification. The central idea is that the same kernel function can be used to define bot a distance measure between points x and the building blocks of the function $f(x)$ to be estimated. As kernel learning can reward sparseness of representation, it can be used to choose within a single round between a large number of competing specifications. Through a "soft-margin" it can abstain from penalizing small errors.

---

[1] A real valued function k that measures the distance between points x_s and x_t within a sample $\{x_1,...,x_S\}$ of size S, is a covariance function or kernel if its Gram-matrix with elements $K_{st}=k(x_s,x_t)$ is positive semidefinite symmetric. For brevity this is also referred to as positive definite.

Third, kernel learning also establishes a unified algorithmic framework that can accommodate semiparametric formulations with a large number of coefficients as well as constraints on coefficients. Despite the highly nonlinear and possibly discontinuous nature of the functions involved, all computations operate within the powerful algorithmic setting of quadratic programming. The analysis starts with a primal program that minimizes the sum of absolute values of the estimation errors incremented by a quadratic regularization term, similar to the one used in ridge regression, which keeps the estimated coefficients bounded. However, this primal problem describes *f(x)* by means of expansions with unknown functions (eigenfunctions). Hence, it cannot be solved directly. The key point is that the associated dual program can, by virtue of the Mercer Theorem – which dates back to 1909 – be expressed in terms of parameters generated by the known kernel functions. This dual program is a quadratic in the coefficients associated to the kernel functions, while the coefficients of the parametric part of the specification appear as Lagrange multipliers. The number of constraints is equal to the number of coefficients in the parametric form, plus one, whereas the primal problem has twice as many constraints as there are observations. Therefore, the dual is also far more convenient computationally.

Finally, kernel mapping offers an encompassing format for poverty mapping. In poverty mapping a regression function *f(x)* on survey data is applied to estimate aggregates, say, at district level by integrating the function over its domain, $\int f(x)dP_i(x)$, where $P_i(x)$ is the frequency of occurrence of x in district i, according to a larger data set, such as a census. Such applications often rule out the use of well founded, theory based parametric forms for *f(x)*, because the census information may not include all the necessary variables in x. The support vector regression of kernel learning is more flexible in incorporating typical covariance patterns, selecting the best functional forms, and truncating of infinite series expansions, than maximum likelihood regressions on polynomial series, and non-parametric regressions such as kernel smoothing whose small sample bias may become objectionable, especially when the surveys are relatively small. Furthermore, in poverty mapping a regression on survey data is to be applied at census level. For this, kernel learning offers a flexible procedure to calculate the necessary redressing weights that can correct for a lack of representativity.

Overview

The paper proceeds as follows. Section 2 gives a brief account of mathematical prerequisites needed to represent a given data set by a function. Section 3 presents the primal and dual quadratic programs of kernel learning, and presents possible confidence bounds. Section 4 describes the kernel learning approach to the problem of classification. Section 5 compares the approach with kernel smoothing and kernel based stochastic optimization. Section 6 discusses application to poverty mapping to include the redressing weights. Section 7 concludes.

Section 2
Representing a data set by a function

2.1 Constructing an orthonormal basis

Suppose that a linear relationship $y(x) = x^T b + b$ holds exactly, while b and b are unknown. We intend to observe the variable $y_S \in R$ at S measurement points with given characteristics $x_S \in R^n$. Knowledge of the measurement points suffices to determine an orthonormal basis (ONB) and to represent the function in terms of this basis. Let r denote the rank of the $S \times n$-matrix X with columns x_S. Now, if X has rank r, $r \le S$, the (b,b)-values in accordance with the observations are characterized by the system of equations $X^T Xb = X^T (y - b)$, and irrespective of the still unknown values of y and b, obey the representation:[2]

$$b = \sum_{i=1}^{r} w_i \sqrt{l_i}\, e_i \tag{2.1}$$

for unknown weights w and where l_i are the r positive eigenvalues of $X^T X$, and e_i are the associated eigenvectors solving $e_i^T X^T X e_i = l_i e_i^T e_i$ and normalized so that $e_i^T e_i = 1$.

Therefore, any point y attainable by the linear function $y(x) = b^T x + b$ can also be accessed (see e.g. Lancaster and Tismenetsky, 1985, ch. 5) directly and "dually" as:[3]

$$y(x) = \sum_{i=1}^{r} w_i y_i(x) + b\,, \tag{2.2}$$

for scaled eigenvalue functions

$$y_i(x) = \sqrt{l_i}\, e_i^T x\,. \tag{2.3}$$

The observations y will determine the values of w and b. This illustrates that a linear function with coefficients (b,b) has a known dual, in this case finite-dimensional, but possibly non-unique representation. The particular representation only depends on the norm chosen (here $X^T X$).

---

[2] Recall from linear algebra that the solution of the system of equations $Kx = b$ can, if it exists, be represented as $x = \sum_i w_i c_i + x_0$ where vectors c_i constitute a basis and w_i are scalars. In fact the summation term is referred to as the kernel (space) of K, or $Ker(K)$. Recall also that a symmetric positive semidefinite $S \times S$ matrix K of rank r can be decomposed into $K = G^T G$, where G is the $r \times S$ matrix of scaled eigenvectors, and $Ker(K) = Ker(G)$.

[3] Throughout the paper, we use summation to express inner products and norms, also when the expression has an infinite number of terms.

Next, suppose that the relationship does not hold exactly. In this case, the unknowns include an error term x_s, and the data impose a constraint:

$$B = \{(w,x) \in (R^r \times R^S) \mid \sum_{i=1}^{r} w_i y_i(x_s) + b + x_s = y_s, s = 1,...,S\} \tag{2.4}$$

Even for *S* much larger than n, the coefficient vector w will not be determined uniquely and the norm that defines the expansion can also serve to select the "best" value, as a solution of:

$$R = \min_{(w,x) \in B} \|(w,x)\|. \tag{2.5}$$

So far, the problem was to select the best coefficient vector from the finite-dimensional normed vector space *B*. That it is possible to represent *S* data points by a polynomial expression of the same order is well known and not very surprising. The key question is however, to establish what happens when the number of points becomes infinite, that is when the empirical distribution approaches the underlying distribution. The major strength of these operations in Hilbert space is that key properties are maintained at infinity. More remarkably, they carry over to nonlinear functions. In fact, the choice of the best values for (w,x) can be interpreted as the selection of the best linear function and for unknown non-linear functions similar relationships hold. Specifically, every bounded non-linear function admits the expansion

$$g(x) = \sum_i w_i \boldsymbol{y}_i(x) + b, \tag{2.6}$$

where w_i are weights, and $y_i(x)$ are orthogonal, scaled eigenfunctions, associated with a positive eigenvalue,[4] all unknown and possibly infinite though countable in number, and *b* an intercept. We do not further elaborate on the properties of these eigenvalues and eigenfunctions, since they are not known and can in general not be identified numerically. However, if a specified norm can be computed to measure the distance between two points, a dual representation can be shown to exist, with unknown coefficients $\boldsymbol{a}$, but known constituent functions, and where the series of coefficients converges. For example, in a space with quadratic norm, any nonlinear function can be represented by the Fourier series as an expansion $g(x) = \sum_i \boldsymbol{a}_i h_i(x)$, for known $h_i(x)$, meaning that on a subset where the quadratic norm can be evaluated, every function can be expressed as a linear combination of terms, in a countable, convergent series. This implies that the selection of the optimal function becomes in some sense like a linear regression problem with coefficients α, albeit that the number of these coefficients is in principle infinite. The Fourier

[4] The scaled eigenfunctions are the product of the root of the eigenvalues of the function induced by the kernel multiplied by the associated eigenfunctions: $y_i(x) = \sqrt{l_i}\, e_i(x)$ for $i = 1,...,r$, where l_i and $e_{i(x)}$ satisfy the eigenvalue equations $\int_X k(x,x') e_i(x) dP(x) = l_i e_i(x')$ and the orthonormality conditions $\int_X e_i(x) e_j(x) dP(x) = d_{ij}$, for $P(x)$ denoting the probability measure of x and d_{ij} equal to unity on the diagonal.

approach has found application to find the optimal policy functions of economic models with known coefficients (e.g. Judd, 2001). Its limitation is, however, that the quadratic norm may not be suitable in many applications, because it cannot represent dissimilarity in a sufficiently flexible way, say, to deal with periodicity, or spatial covariance patterns.

Another entry to the problem of function identification is to start from a given density function $p(x)$ and a loss function $\ell(y,x)$ that is strictly convex in the vector y and integrable w.r.t. this density, and norm coercive, i.e. such that $\ell(y,x) \to \infty$ whenever $\|y\| \to \infty$, i.e. whenever any element of the vector y goes to plus or minus infinity. Then the convex optimization problem

$$R_1 = min_y \int \ell(y,x)\, p(x)\, dx \tag{2.7}$$

yields a unique and bounded optimal choice y^*. Alternatively, suppose that it is possible to choose the best y for every x separately. Then, the optimization identifies the (single valued) function $f(x)$:

$$R_2 = min_{f(x)} \int \ell(f(x),x)\, p(x)\, dx \tag{2.8}$$

i.e. solves a problem in functional space. Clearly, $R_2 \leq R_1$ and a representation of $f(x)$ with a finite number of parameters, produces an intermediate case.

2.2 Function representation

The key step in moving from representation of functions in normed space to kernel learning is that this permits to replace the quadratic norm by any kernel function, and hence, for any given (regularized) kernel function expressing dissimilarity, it yields a particularly transparent representation that is linear in coefficients. Recall that a Banach space is a complete normed space,[5] i.e. for any two elements of the space of objects (not necessarily points in real space) we can compute a vector norm, as a distance measure between the elements measuring the dissimilarity (hence the importance in pattern recognition). A Hilbert space is a complete dot product space: the vector norm is a dot product. In a Regularized Kernel[6] Hilbert Space, the distance measure is a *regularized kernel risk functional,* e.g. a sum of errors incremented by a convex increasing function of choice variables, the regularizer that keeps them in a compact set under minimization of the functional.

[5] A complete vector space is a vector space for which all Cauchy sequences in the space converge.
[6] A differentiable kernel function on the dot product, $k(<x,x'>)$, is positive semidefinite if and only if all the terms of its Taylor expansion are positive (S&S, p. 111).

Like in (2.6), the kernel function defines the eigenvalues and the eigenfunctions associated to positive eigenvalues, and hence the scaled eigenfunctions $y_i(x)$. In fact, this representation can be extended to the semiparametric form:

$$f(x) = g(x) + \sum_j b_j f_j(x) \tag{2.9}$$

where b_j are m unknown coefficients, and $f_j(x)$, $j = 1,...,m$, are known functional forms. The intercept b is subsumed in this list, as $f_j(x) = 1$. Combining with (2.6) yields the representation:

$$f(x) = \sum_{i=1}^{N} w_i y_i(x) + \sum_{j=1}^{m} b_j f_j(x) \tag{2.10}$$

where w_i and $y_i(x)$ are the unknown weights and the scaled eigenfunctions of (2.6) and N is possibly infinite.

The major step to practical application is the Mercer Theorem (S&S, p. 37) which states that the properties of the kernel ensure that it can be decomposed into scaled eigenfunctions

$$\sum_{i=1}^{N} y_i(x) y_i(x') = k(x, x'), \tag{2.11}$$

where the series, if it consists of an infinite number of elements $(N = \infty)$, converges for almost all (x, x'). Note that for linear functions, we have, by construction of the eigenvector (S&S p.584):

$$x' = \sum_{i=1}^{r} l_i \left(\sum_j e_{ij} x_j' \right) e_i \tag{2.12}$$

and, therefore,

$$\begin{aligned} x^T x' &= \sum_{i=1}^{r} l_i \left(\sum_j e_{ij} x_j' \right) \left(\sum_j e_{ij} x_j \right) \\ &= \sum_{i=1}^{r} l_i (e_i^T x')(e_i^T x) \\ &= \sum_{i=1}^{r} y_i(x) y_i(x') \quad . \\ &= k(x, x'), \end{aligned} \tag{2.13}$$

in agreement with Mercer's Theorem.

The other mainstay is the Semiparametric Representer Theorem, which establishes that in the normed space induced by a given regularized kernel, each minimizer f of the regularized empirical risk functional admits a representation of the form

$$f(x) = \sum_{s=1}^{S} a_s k(x_s, x) + \sum_j b_j f_j(x) \tag{2.14}$$

which is the general form of the regression function whose coefficients are to be estimated by kernel learning.

Note that the function inherits all the differentiability properties of the kernel and f-functions, and conversely, a discrete kernel leads to a discrete function. Yet, as is well known from logit-regression, there are ways to represent discrete choice by means of continuous functions, say, as *sign(f(x))*.

2.3 Constructing kernels

The range of possible kernel functions is wide. We briefly review some of those that are most frequently used. Many other examples can be found in Genton (2001).

Polynomial: $k(x,x') = \langle x,x' \rangle^d$, for positive, integer *d*.

Gaussian radial basis. $k(x,x') = exp\left(-\frac{\|x - x'\|}{2s^2} \right)$. The resulting matrix *K* has full rank.

Neural network. $k(x,x') = tanh(k\langle x,x' \rangle + J)$, where $k > 0$ and $J < 0$.[7]

Both the polynomial and the Gaussian radial basis are positive definite (this means that the matrix with element K_{st} is positive semidefinite: $c^T Kc \geq 0$ for all *c*). Real-valued kernels are symmetric. Kernel function are not necessarily continuous but $k(x',x)$ should be integrable w.r.t. *x*. The neural network or sigmoid kernel is not positive definite but it has successfully been used in practice (S&S, p.113). Some applications use conditionally positive definite kernels ($c^T Kc \geq 0$ for all *c* such that $c^T i = 0$), e.g. to allow for translation of origin.

Users can tailor kernels to their specific purposes building on the following properties (see Genton, 2001, S&S p. 408):

(a) The weighted sum of two kernels is a kernel.
(b) The product of two kernels is a kernel
(c) The exponent of a kernel is a kernel.
(d) If *g* is a real valued scalar function, then $k(x,x') = g(x)g(x')$ is a kernel.
(e) If *g* is a R^n-valued function and *h* a kernel on $R^n \times R^n$, then $k(x,x') = h(g(x),g(x'))$ is a kernel.

Note that by (b) and (d), we can use the kernel to scale the variables i.e. use $k(x,x') = g(x)h(x,x')g(x')$. Recall, however, that the kernel enters the dual program only. A

[7] Recall that $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ has a sigmoid shape, range $(-1,1)$, that it passes through the origin.

high value will tend to attribute a larger weight and hence to yield a lower α-coefficient on the observation concerned, because of the lower "information content" of the observation. To stress the covariance properties of the function, one might consider the following kernel. Let *h* be a real valued function on *X*, that is positive with minimum at zero (a variance function), then

$$k(x,x') = \frac{1}{4}[h(x+x') - h(x-x')] \tag{2.15}$$

is a kernel, and the plus-term reflects a global weight (distance from origin) while the minus-term reflects distance. The motivation is that for two random variables *x* and x':

$$cov(x,x') = \frac{1}{4}[var(x+x') - var(x-x')]\,. \tag{2.16}$$

In practice, preference is for kernels that have zero value outside a given domain (bounded support), since this generates sparse matrices *K*, which is essential to keep calculations tractable in the mathematical programs of the next section. Note that it is not possible simply to disregard values in *K* that fall below some given threshold, since the resulting matrix will, in general, not be positive semidefinite. Using (b), a compact support can be obtained through multiplication of the kernel function by

$max(1 - \frac{\|x-x'\|}{q}, 0)$, where q is a window size.

In the applications below the kernel only needs to be conditionally semipositive definite (cspd), $\sum_s \sum_t c_s c_t k(x_s, x_t) \geq 0$ for all c_s such that $\sum_s c_s = 0$. This makes it possible to define a wider class such as $k(x,x') = -\|x-x'\|^b$ for $0 \leq b \leq 2$. It also illustrates that the kernel formulation covers the case where the kernel is interpreted as the negative of the distance. Fields of application are often characterized by their choice of kernel function. For example, in road transport, one could use the shortest road distance to define $k(x,x')$, or the least cost route, as the solution of a cost-minimizing optimal routing problem, with a nonnegative, possibly discrete decision variable $v \in V$:

$$k(x,x') = max_{(v_1,\ldots,v_n) \in V} \sum_h v_h k_h(x,x')\,, \tag{2.17}$$

where the kernel function could be the negative of the cost using vehicle *h*, to transform *x* into x', and vice versa, and by (a) above, yields a cspd Gram matrix. Also, given a fixed set of observations, it is possible to optimize transport over several points simultaneously. The specification illustrates that symmetry is the major restriction, as in fields other than

transportation (e.g. health, education, demography) few transformations are fully reversible, and distance presumably is a better metaphor than transport cost, since after all, the kernel measures comparability.

Finally, specification (2.15) indicates that if the kernel has zero as its lower limit, the function value at points far out of sample will be determined by the parametric part, whereas if, say, the kernel is the negative of the distance measure, it becomes minus infinity, and obviously dominates.

Section 3
Function estimation

3.1 Support vector regression and quadratic programming

The next step is to estimate the α- and b-coefficients of (2.15). For this, kernel learning formulates a primal quadratic program of support vector regression that minimizes a specified regularized risk functional. The regression error consists of two parts, a soft-margin ε common across all observations and a specific error ξ_s, ξ_s^* for positive and negative deviations, respectively. Similarly, to penalize absolute values, coefficients are expressed as difference between two nonnegative values ($\alpha - \alpha^*$, and $b - b^*$, respectively). The program more severely penalizes the sum of absolute values of the specific errors than the deviations within the soft margin. In addition, its objective includes, as regularization terms to keep the coefficient of expansion (2.15) bounded, the squared quadratic norm of the weights w, with a fixed error penalization factor ρ, and the absolute value of the coefficients of the parametric part:

$$R = \min_{\xi_s, \xi_s^* \geq 0, all\, s; \beta_j, \beta_j^* \geq 0, \varepsilon \geq 0, w} \frac{1}{2}\|w\|^2 + \sum_j \kappa_j (\beta_j + \beta_j^*) + V\varepsilon + \rho \sum_s (\xi_s + \xi_s^*)$$

subject to

$$\begin{aligned} y_s - \sum_i w_i y_i(x_s) - \sum_j (\beta_j - \beta_j^*) f_j(x_s) &\leq \varepsilon + \xi_s \qquad (\alpha_s) \\ \sum_i w_i y_i(x_s) + \sum_j (\beta_j - \beta_j^*) f_j(x_s) - y_s &\leq \varepsilon + \xi_s^* \qquad (\alpha_s^*), \end{aligned} \tag{3.1}$$

for given $\rho = C/S$, and $V = \nu C$, where C is a scalar that controls the weight on the error, relative to regularization, and $\kappa_j = 0$ in case j refers to the intercept i.e. $f_j(x_s) = 1$, and $\kappa_j = 1$ otherwise (S&S, p. 262-263).[8] Clearly, these values could differ across j. The observations for with the constraints hold with equality are support vectors and this type of regression is called support vector or SV-regression because the multipliers are determined by these active constraints only.

The higher the value C, the more the emphasis lies on the reduction of errors in the first term of the objective. The fraction ν is also an upper bound on the fraction of observations for which positive errors are accepted, as opposed to the observations for which penalization is avoided by adjustment of ε. The empirical risk is the regularized sum of deviations $R_{emp} = R/\rho$ and is nonnegative.

This semiparametric approach avoids the arbitrariness of stepwise procedures that estimate the non-parametric part from the errors of a parametric model. The formulation seeks a

[8] S&S occasionally use C instead of ρ, but we maintain ρ to keep the objective bounded for $S \to \infty$.

compromise between efficiency, as expressed in the small sum of absolute values of the regression errors, and sparseness of the representation, as expressed by the regularization terms on w, and b. A high value of the regularization parameter r favors sparseness. A further justification of the regularization, is that to any regularized kernel corresponds a pure kernel function that selects the same function.

The difficulty in (3.1) is, obviously, that it poses the problem in functional space, since $y_i(x)$ is given in the optimization problem, and yet unknown and, moreover, the number of such terms might be infinite (even though by the Mercer Theorem the expression is converging for almost all pairs (x_s, x_t)).[9] The main step is that the α- and b-coefficients are readily derived from the (Wolfe) dual quadratic program (see Appendix for a derivation):

$$R = \min_{a_s, a_s^* \geq 0} \frac{1}{2}\sum_s \sum_t (a_s - a_s^*) k(x_s, x_t)(a_t - a_t^*) - \sum_s y_s (a_s - a_s^*)$$

subject to (3.2)

$$\sum_s (a_s - a_s^*) f_j(x_s) \leq k_j \qquad (b_j^*)$$

$$\sum_s (a_s - a_s^*) f_j(x_s) \geq -k_j \qquad (b_j)$$

$$a_s \leq r$$

$$a_s^* \leq r$$

$$\sum_s (a_s^* + a_s) \leq V \qquad (e)$$

where we note that for $\kappa_j = 0$, the first two inequalities collapse into an equality, with multiplier $\beta_j^* - \beta_j = \gamma_j$. This amounts to dropping b and b^* from the objective in (3.1), and may therefore, avoid a bias in these coefficients. As long as κ_j is positive, they will remain bounded, and if it is zero, boundedness is ensured if constraint qualification holds, i.e. if the rows of matrix $f_j(x_s)$ are linearly independent across j.

Furthermore, note, first, that a_s and a_s^* cannot exceed vC. Hence, the upper bounds will be ineffective for $v < 1/S$, but in this case, there is hardly any emphasis on fitting a function since ε can adjust almost freely. Secondly, the upper bounds mitigate the effect of outliers on the

[9] The critical requirement is that the kernel matrix should be conditionally positive semidefinite. Sufficient conditions were mentioned in the previous section. In addition, to check whether a specific matrix meets the requirement, one may solve a quadratic program whose objective is the quadratic term in (3.2), and with a normalization $\sum_s a_s + a_s^* = 1$, while $\sum_s a_s - a_s^* = 0$. If the value R is negative in the optimum, the condition is violated.

estimation. Thirdly, whenever the regression function has an intercept, i.e. whenever $f_j(x) = 1$ for some j, we may use a conditionally (semi) positive definite kernel. Fourthly, since the regularization penalizes the absolute value of α-coefficients, these coefficients will be zero for all zero terms of the kernel. For example, if the kernel matrix was purely diagonal, the expansion would only count one non-zero coefficient at most. By the same token, it will in general be impossible to obtain a perfect fit, even with $\nu = 1$. This also implies that a sparse kernel matrix is less well equipped for evaluation of the function at points other than the observations. Moreover, the β-coefficients of the parametric part of the function can in this respect make good for a lack of α-coefficients, and vice versa, but the relationship is complex, and with a diagonal kernel matrix the parametric part cannot be identified.

The main point is that it was possible to eliminate all primal variables from the dual program, because the primal is linear in the error variables and, because, by Mercer's equality, the left-hand side of (2.15) that appears in the dual's objective can be replaced by the known kernel function.

As an alternative to the dual quadratic program, we may also consider the linear program, by substituting the explicit representation in the primal:

$$R = \min_{a_s, a_s^* \geq 0; b_s, b_s^* \geq 0; e \geq 0; x_s, x_s^* \geq 0, all s;} \sum_s (a_s + a_s^*) + \sum_j k_j (b_j + b_j^*) + Ve + r \sum_s (x_s + x_s^*)$$

subject to (3.3)

$$y_s - \sum_t k(x_s, x_t)(a_t - a_t^*) + \sum_j f_j(x_s)(b_j - b_j^*) \leq e + x_s$$

$$\sum_t k(x_s, x_t)(a_t - a_t^*) + \sum_j f_j(x_s)(b_j - b_j^*) - y_s \leq e + x_s^*$$

for given $r = C/S$, and $V = \nu C$, as in (3.1). These programs need further comment and allow for several modifications.

Size. Note that the dimensionality of this problem is far larger than that of (3.2). Here we have $4S + 2m + 1$ variables, and $2S$ constraints, while (3.2) counts $2S$ variables and $2m + 1$ constraints, which reduce to $m + 1$ if $k = 0$.

Soft margin. The soft margin ε is recovered from (3.2) as the Lagrange multiplier on the νC-constraint. The optimization penalizes nonzero α-coefficients, and the smaller ν, the fewer the number of coefficients will be, but the poorer the fit. Conversely, the higher C, the stronger the emphasis on a good fit. It is possible to modulate the margin , replacing ε by $\eta_s \varepsilon$ in the constraints, with given nonnegative factors η_s adding up to S. In the dual program these weights appear on the left-hand side of the νC-constraint. The higher the factor, the lower the α-weight, and hence the more modest the role of this observation in the expansion.

Observations are iid. The error terms in (3.1) are taken to be iid. Despite the possible interpretation of the kernel matrix $K_{st} = k(x_s, x_t)$ as an expression of covariance among coefficients, the specification assumes that observations are independent and identically distributed (iid). This is an apparent limitation but it may be argued that the non-parametric part of the functional form can represent and hence absorb all covariance, and the Representer Theorem establishes that this provides a perfectly flexible representation of the data. For example, time-series can be represented by including time as an explanatory variable and possibly by allowing for periodicity in the kernel.

Ridge regression by least squares. The regressions minimize the sum of absoluted values of errors $\boldsymbol{r}\sum_s(\boldsymbol{x}_s + \boldsymbol{x}_s^*)$ incremented by a regularization term, quadratic in program (3.1) and linear in (3.3). The standard format of ridge regression by least squares of maximum likelihood with normally and iid distributed error is obtained if in the linear program (3.3) the summation of absolute values is replaced by $\boldsymbol{r}\sum_s(\boldsymbol{x}_s - \boldsymbol{x}_s^*)^2$ while the soft margin $\boldsymbol{e}$ is being dropped, so that each pair of inequality constraints collapses into an equality. Note, however, that this does not reduce to ordinary least squares because the information matrix $\tilde{X}^T\tilde{X}$, where $\tilde{X}$ is the matrix of all independent variables, is structurally singular. Yet, the conversion illustrates that the fundament of kernel learning is the use of the semiparametric representer theorem, while the choice of format for parameter estimation is a matter of convenience more than of principle.

Asymmetries. A further key requirement on the quadratic program is that the kernel function should be symmetric semipositive definite. Some applications may contain inherent asymmetries, so as to reflect direction, say, of time, spatial flows, or social hierarchies, where upstream observations can have a downstream effect but not the other way around. In view of the symmetry requirement it is not possible to impose this in kernel learning, writing, say, $k(x_s, x_t) = 0$, if $x_h(s) < x_h(t)$ and $k(x_t, x_s) > 0$ otherwise. However, in many cases it is possible to transform the series of the dependent variable y into a form that eliminates the asymmetry. For example, suppose that the model is postulated to be

$$\vec{y} = L\vec{y} + \vec{f} \tag{3.6}$$

where we write $\vec{x}$ to denote the vector with elements x_s, and $\vec{f}$ for the vector with elements $f(x_s)$, and where the $S \times S$ matrix L is lower triangular, a known function of x with zero on its diagonal, to reflect the recursivity. We can now transform the y-data accordingly by $\tilde{y} = (I - L)\vec{y}$ and estimate $\vec{f} = K\boldsymbol{a} + F\boldsymbol{b}$, in the usual way, where $\boldsymbol{a} = \boldsymbol{a} - \boldsymbol{a}^*$. The result can be inserted in (3.6), and in view of the recursive structure, the elements of $\vec{y}$ can be recovered one by one. It is also possible to eliminate y on the right hand side and obtain the full form:

$$\vec{y} = (I - L)^T \tilde{y} = (I - L)^T Ka + (I - L)^T Fb \tag{3.7}$$

Note that if the Kernel matrix itself is defined as $K = (I - L)(I - L)^T$, this simplifies further to:

$$\vec{y} = (I - L)^T (a + Fb). \tag{3.8}$$

Note that this representation is as if we had used an asymmetric kernel matrix $\tilde{K} = (I - L^T)$ with a special structure on the parametric part. We mention that this formulation has the disadvantage that it can only be evaluated on the full grid of points of the sample itself.

Autoregression. It is also possible to represent autoregression directly by specifying a form such as $y_t = f(z_t, z_{t-1}, y_{t-1}, t) + z_t$ for the model itself, possibly with the lagged variables appearing in the parametric part only, and treat the arguments of *f* as the independent variable and the error as iid. Time series analysts could object that this neglects the endogeneity of y_{t-1} and the autocorrelation z_t. Yet, in kernel learning, only the function is unknown, while all data are exogenous and may therefore appear as right hand side variables.

Frontier estimation. If the primal problem contains lower or upper bounds only, it turns into a frontier estimation that determines the hull of points, and for a suitable specification of kernel, this hull can be made convex.

Constraints on b. It is straightforward to impose linear restrictions on b Suppose that the constraints $Db \le d$ are imposed on the coefficients b, and that strict inequality is feasible. These constraints can be inserted in (3.1), with non-negative Lagrange multipliers l. The associated dual (3.2) can be extended by incrementing the objective with $\sum_h l_h d_h$, while the constraints become

$$-k_j \le \sum_h l_h D_{hj} - \sum_s (a_s - a_s^*) f_j(x_s) \le k_j. \tag{3.9}$$

No regression with intercept on a constant. If the regression function has a constant $f_1(x_s) = 1$, the program cannot be used to estimate the coefficients of an implicit function, i.e. for a constant left-hand side variable, since all a's will then be set to zero.

GAMS implementation. Numerical implementation of the quadratic program in GAMS (Brooke et al. 1998) has proved relatively straightforward, since it essentially require convex quadratic programming only. The package proved especially suited because of its user friendliness in

representing different parametric forms and kernel functions, and its capability to process a sparse representation of the kernel matrix.

Finally, some limitations may be mentioned. The Representer theorem ensures that any finite data set can be represented with perfect accuracy by any kernel function, with at most as many parameters as there are observations. It does not establish how stable these coefficients are when the data set is extended. Like any flexible form, the kernel representation uses up all degrees of freedom, and clearly, in a pure regression context it would be impossible to identify α and b jointly. Conversely, since the kernel spans the full range of the function, variations in x, can easily lead to wild variation in f, in any direction. Kernel learning addresses this inevitable tension between goodness of fit and significance (stability of prediction), that has its equivalent in the estimation of the covariance matrix in maximum likelihood methods in two ways. First, via the regularization coefficient C, and the soft margin ν it introduces in the quadratic program (3.2) an explicit penalization on inessential coefficients, like under ridge regression.[10] This will ensure that there are less nonzero coefficients than observations. The C-coefficients essentially bound individual α coefficients, while the soft-margin factor ν introduces competition among coefficient values. Both allows the γ-coefficients to become larger. Second, through its functional form (e.g. window size), the kernel function controls the dependence (covariance) across observations. Whereas small window size leads to a pure diagonal form that may fit one parameter to every observation, non-zero off diagonal terms specify connections, and through it limit the influence of outliers.

3.2 Test statistics

Estimation needs test statistics to assess the quality of the results. This is a somewhat technical subject, and since applications generally rely on pre-established formula, we only present four of these.

The first statistic simply counts errors. For given precision ε, we can calculate the ratio of effective bounds over the total number of observations, as the probability of error (S&S, p. 369).

A second statistic defines the probability of loss for the Gaussian radial basis function (RBF) kernel (S&S, p. 364) that is often used in kernel density regression. It is calculated as the probability of a particular deviation between empirical and true risk exceeding a specified value:

---

[10] It would be possible to estimate the functional form (2.11) by linear (ridge) regression, minimizing the sum of square errors ξ, and the sum of square coefficients multiplied by the regularization factor ρ. This yields a least squares vector $q = \left(Z^T Z + rI\right)^{-1} Z^T y$, where Z is the matrix of independent variables, and θ the vector of coefficients. Note that the regularization maintains nonsingularity of the matrix to be inverted, but also causes a bias in the coefficient. Since the dimension of the matrix to be inverted is usually large, it is impractical to evaluate the coefficients in this way. Linear program (3.3) can be interpreted as a ridge regression that minimizes the sum of absolute values of error terms, and has a soft margin with penalty factor V.

$$P\{|R_{emp} - R_{true}| < \bar{e} + h\} \le 2\exp\left(-\frac{S}{2}\left(\frac{\bar{e}}{M}\right)^2\left(1+\frac{2}{rM}\right)^{-2}\right), \tag{3.10}$$

where M denotes an upper bound on the loss and $h = \frac{2}{rM}$. The right hand side in the probability is called confidence term, or capacity term. It gives a percentage indication of how wrong the estimate might be, and right hand side of the outer inequality assigns a probability to this loss. We observe that, for small r, this bound becomes meaningless.

Finally, likelihood ratios, measured as ratios of empirical risk, can be used to assess the cost of particular restrictions on the quadratic program (3.2). For example, one may consider the tightening of upper bounds on α-coefficients in the dual program. This raises the objective as well as the empirical risk. Conversely, dropping some term $f_j(x)$ reduces the objective, since it relaxes restrictions on α.

3.3 Representation on a reduced set

The kernel learning approach has the drawback that the number of α- coefficients may become too large, and that it is difficult to formulate adequate test statistics to evaluate their stability. Moreover, the number of coefficients was seen to depend on the regularization constant C and the precision factor v. Recall also that the dual quadratic program may set many of the α-coefficients at their upper bound. It would seem natural to replace all kernel functions with this common coefficient by a smaller set of representative observations.

To arrive at more compact and stable formulations, current research in kernel learning proposes to include a final stage after the solution of the quadratic program, that treats the α-coefficients themselves as data and looks for a representation in smaller dimensions with fewer nonzero coefficients p_s and p_s^* (S&S, p. 554) The aim is to find these coefficients and data points i.e. to minimize as in ridge regression the regularized objective, without further constraints:

$$min_{p_t, p_t^* \ge 0, d} \ \frac{1}{2}\|d\|^2 + \Sigma_t (p_t + p_t^*) \tag{3.11}$$

for

$$d_s = \left(\Sigma_t (a_t - a_t^*) k(x_t, x_s) + \Sigma_j b_j f_j(x_s)\right) - \left(\Sigma_t (p_t - p_t^*) k(x_t, x_s) + \Sigma_j b_j f_j(x_s)\right)$$

or, since the b-terms drop out:

$$d_s = c_s - \Sigma_t (\beta_t - \beta_t^*) k(x_t, x_s), \tag{3.12}$$

where $c_s = \sum_t (\alpha_t - \alpha_t^*) k(x_t, x_s)$ serve as observations on the dependent variable. Alternatively, we can apply kernel learning, which in the absence of a soft margin amounts to considering the dual program associated to the primal problem:

$$R = min_{x_s, x_s^* \geq 0, alls; w} \frac{1}{2}\|w\|^2 + r\sum_s (x_s + x_s^*)$$

subject to (3.13)

$$c_s - \sum_i w_i y_i(x_s) \leq x_s \qquad (p_s)$$

$$\sum_i w_i y_i(x_s) - c_s \leq x_s^* \qquad (p_s^*),$$

S&S report that practical experience suggests to use (3.12) or (3.13) only to identify the subset S_1 of points *t* with a nonzero β-coefficients. The last step of the procedure is then to re-estimate the β-values with $r = 0$ and kernels evaluated at these points only. It appears that this amounts to ordinary least squares.

3.4 Clustering

A related question is to group the data into clusters whose members are similar not with respect to a known measure but in terms of a function to be estimated. In poverty mapping this permits to characterize groups of respondents according to the similarity of their profiles, rather than of their poverty. For this, one may consider observations on *x* only, and enclose the data within spheres of minimal radius *r*, with a soft margin (Ben-Hur et al., 2001). For the scalar *c* denoting the central *f*-value, we may write:

$$min_{x_s, r^2 \geq 0, c} r^2 + r\sum_s x_s$$

subject to (3.14)

$$(f(x_s) - c)^2 \leq r^2 + x_s \qquad (p_s)$$

where we optimize over the function *f* as well for a given kernel function *k*. The first-order conditions imply that (i) $\sum_s \beta_s = 1$ for positive radius; (ii) $c = \sum_s p_s f(x_s)$; (iii) $p_s + h_s = r$, with $h_s = 0$ whenever $x_s > 0$. Hence, the dual program is:

$$
\begin{aligned}
&\min_{p_s \geq 0} \Sigma_s \Sigma_t p_s K_{st} p_t - \Sigma_s K_{ss} p_s \\
&\textit{subject to} \\
&\Sigma_s p_s = 1 \qquad (r^2) \\
&p_s \leq r \qquad (x_s)
\end{aligned}
\tag{3.15}
$$

while the function has the expansion:

$$f(x) = \Sigma_s p_s k(x_s, x) \tag{3.16}$$

and the radius *r* can be recovered as the distance of a point x_t with non zero α-coefficient (a support vector) to the center and form the boundaries of the set of clusters: $r^2 = (f(x_s) - c)^2$ for any *s* such that $p_s > 0$.

Within this sphere, a cluster can be defined as a subset of points that can be connected by line segments which do not exit the sphere, in the sense that every evaluation of convex combinations of two points of the (highly nonlinear) function $f(x)$ remains within the sphere: $(f(x) - c)^2 \leq r^2$. This can be represented in an $S \times S$-dimensional adjacency matrix whose elements have value of unity if two observations are connected by a path that does not leave the sphere and zero otherwise: $x(m) = mx_s + (1-m)x_t$, for $m \in [0,1]$. The larger the window size of the kernel function, the smoother the distance measure $(f(x(l)) - c)^2$, the smaller the number of clusters. We also observe that the kernel matrix itself already provides a non-parametric clustering, from which an adjacency matrix could be constructed.

Programs (3.14)-(3.15) illustrate that there are several variations to the quadratic programs (3.1)-(3.2), for which the Mercer theorem applies. We also note that it would be possible to substitute (3.16) into (3.14) and solve the resulting program directly, but this is far more difficult, because of the quadratic constraint that is far more difficult to handle than a quadratic objective, and the larger number of variables ($2S + 2$ as opposed to *S*).

Section 4
Classification

In practice, kernel learning is mainly used in the context of classification, where the dependent variable points to a particular combination of attributes. Through association to this correct classification with input data, a training data set is developed, on the basis of which it becomes possible to develop a function that can make expert judgements. Poverty analysis counts many instances where the expert's, judgement on the severity and nature of the problem as laid down in, say, patient records, could be related to more quantitative data on age, gender, educational status and location. In this case, the estimation can be interpreted as a learning device, to teach the regression function how to recognize poverty at the level of the individual.

In fact, with a dependent variable y_s equal to zero or unity, the dual quadratic program would estimate a probability of finding $y_s = 1$ (e.g. Greene, 1997). The associated discrete choice could be that one decides that at a given point *x*, the value is unity if this probability exceeds one half, indicating that it is more probable than not. Interestingly, in the field of kernel learning, one generally follows a different approach that has proved successful. Recall that in limited dependent regression such as probit estimation, with *0* and *1*, or in regression (3.1), the observations with *0* count as heavily as those with *1*. It represents non-occurrence by -1 and occurrence by *1* and postulates:

$$y = f(x) = sgn(\sum_i w_i y_i(x) + b) \tag{4.1}$$

and this function can be estimated for *S* observations from:

$$\begin{aligned} &min_{w,b} \frac{1}{2}\|w\|^2 \\ &subject\ to \\ &(\sum_i w_i y_i(x_s) + b) y_s \geq 1, all\ s \qquad (a_s) \end{aligned} \tag{4.2}$$

Hence only the points where the constraint is binding matter. All other points will have zero coefficients. The associated dual is:

$$\begin{aligned} &min_{a_s \geq 0} \frac{1}{2}\sum_s \sum_t a_s a_t y_s y_t k(x_s, x_t) - \sum_s a_s \\ &subject\ to \\ &-\sum_s a_s y_s = 0 \qquad (b) \end{aligned} \tag{4.3}$$

The advantage of treating classification as distinct from ordinary regression is that it only needs positive coefficients, and a single constraint, and that there is no need for regularization.[11] The resulting estimated classification function is:

$$\tilde{f}(x) = sgn(\sum_s a_s y_s k(x_s, x) + b) \tag{4.4}$$

With respect to test statistics, the maximum probability of error is the expected probability of test error (over repeated samples), $P_h = \sum_s t_s$, $s \in T_h$, where T_h is the h-th training set and $t_s = max(y_s \tilde{f}(x_s), 0)$.

Typically, classification is multi-dimensional. In fact, kernel learning has made its proofs in this field, as a tool to recognize individual words from a dictionary, on the basis of handwritten or of vocal input. To train the machine, the user scans in a handwritten version of a given text, or reads this text aloud. In this way the machine receives multiple input x, for every given cell of the classification. Eventually, the machine can reverse the operation, and on the basis of the estimated function, point to a single cell. Multiple classifications (S&S, p. 211) can be obtained from:

$$c = arg\,max_c\, g^c(x)$$

for (4.5)

$$g^c(x) = \sum_s a_s^c y_s k(x_s, x) + b^c,$$

and where every regression function j is a binary classifier of one-against-the-rest, or winner-takes-all. Hence, for every item one runs a single regression, and in the end the one that offers the smallest expected probability of misclassification is chosen. This approach is to some extent heuristic because of the separate estimation for every c, but its advantage as opposed to, say, ordered logit estimation, is that it allows for a different specification for different classes, and thus for arbitrarily many classes, which is important because the number of classes becomes massive as soon as the expert's report has any detail.

Despite its statistical shortcomings, the approach seems to work well in practice, where it appears on the one hand that little is gained from simultaneous estimation (S&S, p.214), and on the other hand that the choice of kernel matters but the window size is often more critical (S&S, p.216).

[11] Here also, the kernel only has to be conditionally positive definite.

Section 5
Applications with kernel smoothing and stochastic optimization

5.1 Kernel smoothing

Remarkably, kernel smoothing, also referred to as kernel density regression, which has become common in econometrics (Haerdle, 1993), follows a very different approach. It relies on a kernel of the form $k((x_s - x_t)/q)$, where q is a windows size or bandwidth, that depends on the number of observations, and goes to zero for $S \to \infty$:[12]

$$f_q^S(x) = \frac{1}{S} y_s k((x_s - x)/q), \tag{5.1}$$

where we use the superscript *S*, and the subscript q to emphasize the dependence of the function on both parameters. Thus, the form agrees with a non-parametric version of (2.6), with coefficients $(a_s - a_s^*) = y_s / S$ and $b_j = 0$. Indeed, in (3.2) it is possible to drop a^* and impose a restriction by the restriction $(1-e)y_s / S \le a_s \le (1+e)y_s / S$. For $S \to \infty$ and fixed q, and supposing that *f(x)* is continuous almost everywhere when *k* is positive, the estimator (5.1) converges to:

$$f_q(x) = \int f(x+z)k(z/q)dz, \tag{5.2}$$

and for $q \downarrow 0$ it converges to $y(x)$.[13] In fact, the function *f(x)* estimated by kernel learning also depends on *S* and q. For general kernels, and if *f(x)* is continuous almost everywhere on *P*, the kernel smoothed function is

$$\tilde{f}(x) = \int_X k(x', x) f(x') dP(x'). \tag{5.3}$$

This function also inherits the differentiability properties of the kernel. Furthermore, let l_i^S denote the *i*-th smallest eigenvalue of the iid sample of size *S*, K^S the associated Gram-matrix, and *P(x)* the probability distribution on a compact domain $X \subset R^n$. Since

[12] Kernel smoothing generally applies a division by $\sum_S k(x_S, x)/S$, and interprets the kernel weights as probabilities. The deviation of this denominator from unity also measures the small sample bias and creates some correction for it. For large *S*, this denominator converges to unity if the kernel is a density, but the ratio creates some problems w.r.t. consistency and we therefore discard the division.

[13] The convergence property of smoothing is weaker, since it is pointwise, whereas the consistency of kernel learning can be established in functional space, i.e. w.r.t. to integrals of absolute values of deviations between the estimated and the true function.

$\frac{1}{S}\sum_{i=1}^{S} l_i^S = \frac{1}{S} tr(K^S)$ and $l_i^S \geq 0$, and the kernel is integrable, we have $lim_{S\to\infty} \frac{1}{S}\sum_{i=1}^{S} l_i^S = \int_X k(x,x)dP(x) = Const.$, by the law of large numbers.This shows that if samples are iid, the mean of the eigenvalues of an integrable psd kernel converges to a nonnegative constant, and if this constant is positive, as is the case for a radial basis kernel, the sum of eigenvalues goes to infinity for $S \to \infty$.

We note that if we concentrate on a fixed number r of largest eigenvalues, then for an infinite number of observations obeying *P(x)*, the kernel smoothed eigenfunction coincides with its non-smoothed counterpart $e_i(x)$, for $i = 1,...,r$. Specifically, consider the data sample of size $S = r$ with observations x_s. For every sample we can evaluate the eigenvectors e_i, ranked in decreasing order of the eigenvalue, with elements e_{is} and estimate the kernel smoothed eigenvector function:

$$e_i^S(x) = \frac{1}{S}\sum_s k(x_s, x)e_{is} \tag{5.4}$$

Hence, for a finite sample it is possible to estimate these eigenfunctions by kernel smoothing and the consistency properties of kernel smoothed functions, for constant, positive window-size, can be used to prove the existence of these *r* eigenfunctions for $S \to \infty$. Clearly, it is also possible to apply kernel regression (3.2) to estimate this function, with e_{is} as dependent variable.

The main difference of kernel smoothing from the support vector regression of kernel learning is that the estimate from kernel learning is based on minimization of an empirical risk functional whereas (5.1) only applies a prespecified smoothing which makes it less efficient, but also far less costly, since it avoids solving the quadratic program altogether. We also note that there is no basic difficulty in allowing for asymmetric kernels in this context, even though the theory assumes symmetry. We observe that for regression on binary variables, the kernel smoothing (5.1) would only take into consideration the non-zero values. Hence, as in (4.1), it would be more appropriate to represent non-occurrence by -1, and derive the probability of occurrence as $P_q^S(x) = f_q^S(x)/2 + 1/2$.

5.2 Risk minimization by stochastic quasigradient optimization

When the number of observations grows large, and the kernel value does not vanish quickly between them, the kernel matrix can become too large for incorporation in the dual quadratic program. In such a situation there is a need for iteration over alternative quadratic programs. In principle a wide array of decomposition techniques can be used for this. These methods optimize over a subset S_1 of observations while keeping the other α-values fixed. Given these optimal

values, one subsequently chooses another subset, S_2, and so on. Since the earlier optimum remains feasible, the new risk value R will be higher, and the procedure will eventually converge to the optimum of (3.2), see e.g. Collobert and Bengio (2001).

Alternatively, one may consider the use of stochastic techniques. Stochastic quasi-gradient optimization has been used to solve optimization problems with integral functions such as risk functionals, or with a large number of variables (Ermoliev, 1988). Application of SQG-principles would amount to treating all data g^h entering batch h of the quadratic program, as being drawn at random from an empirical or otherwise basic distribution, and in every batch, to adjust the coefficient in a smoothed way, until convergence. We note, that this approach is not suitable to estimate $\{a_s, a_s^*\}$ since these coefficients fully depend on the specificity of every sample. Hence, it only appropriate for estimation of the parametric part, which, as was seen in (3.11) above, might include minimal kernel expansions.

The application would consider a series of minimizations indexed h, that include in (3.1) the given value b, from the previous iteration, and the data set (x, y):

$$R(b; x, y) = \min_{x_s, x_s^* \geq 0, all\ s;\ b_j, b_j^* \geq 0\ e \geq 0, w} \frac{1}{2}\|w\|^2 + \sum_j k_j (b_j + b_j^*) + Ve + r\sum_s (x_s + x_s^*)$$

$$subject\ to \tag{5.5}$$

$$y_s - \sum_i w_i y_i(x_s) - \sum_j (b_j - b_j^* + b_j) f_j(x_s) \leq e + x_s \qquad (a_s)$$

$$\sum_i w_i y_i(x_s) + \sum_j (b_j - b_j^* + b_j) f_j(x_s) - y_s \leq e + x_s^* \qquad (a_s^*),$$

while penalizing by a now positive k_j the deviations from b_j. In terms of the dual program (3.2), this amounts to dealing with the adjusted observations: $\tilde{y}_s = y_s - \sum_j b_j f_j(x_s)$, that can be interpreted as the so far unexplained part of the parametric regression. The adjustment of b can proceed so as to solve the problem:

$$\min_b \int R(b; x, y)\, dP(x, y) \tag{5.6}$$

where $P(x, y)$ is the empirical probability distribution of the samples. We note that the dual of (5.5) shows that R is convex in b. If we denote the h-th random sample by (x^h, y^h), this amounts to solving:

$$\min_b \lim_{H \to \infty} \frac{1}{H} \sum_{h=1}^{H} R(b; x^h, y^h). \tag{5.7}$$

Now, the global optimum of (5.6) can be approximated iteratively by the SQG-process:

$$b^{h+1} = b^h - s_h g^h, \qquad h = 1,2,... \tag{5.8}$$

where $g_j^h = \sum_s (\alpha_s^h - \alpha_s^{*h}) f_j(x_s^h)$, is the subgradient of R w.r.t. b for the given data set x^h, y^h, and hence an SQG, and s_h is a stepsize, that eventually should not exceed $1/h$. To initiate the process, it seems natural to use $g_j^h = (b_J^{*h} - b_j^h)$ but eventually the SQG should be used. Iteration (5.8) almost surely converges to a global optimum of (5.6).

Finally, we mention that Ermoliev et al. (2002) consider a different approach that avoids the curse of dimensionality of quadratic programming, by incorporating a semiparametric form in a stochastic quasigradient (SQG) framework, where the kernel is used to represent the density function.

Section 6
Poverty mapping

This section discusses two connections of kernel based methods with poverty mapping. The first presents a kernel smoothing approach to derive regression weights for support vector regression of a poverty indicator. The second shows how the kernel function can be used to test the adequacy of the matching between survey and census distributions.

6.1 Redressing survey data for poverty mapping

As mentioned in the introduction, poverty maps can be constructed by applying Monte Carlo integration to the regression function *f(x)* over a census with a probability distribution $G(x)$, and district sets X_g for geographical districts *g*. Kernel learning also applies to this case. Here we consider SV-regression but classification problems can be dealt with in a similar way. To account for the census data, one postulates that the errors in the regularized empirical risk functional are to be weighted by their probability of occurrence in the census. For this, the kernel function can be used as well. It serves to attribute a redressing weight[14]

$$N_s = \int k(x_s, x)\, dG(x) \tag{6.1}$$

to survey observations. In fact, the current practice (e.g. Hentschel et al., 2000) is to conduct the regression on the survey, independently of the census it is subsequently applied to. This amounts to neglecting potential heteroskedasticity and may bias the the estimates. To incorporate the source of heteroskedasticity, we modify the SV-regression problem (3.1) to:

$$R = \min_{w; b_j, b_j^* \geq 0, e \geq 0, x_s x_s^* \geq 0;} \frac{1}{2}\|w\|^2 + \sum_j (b_j + b_j^*) + Ve + r \sum_s N_s (x_s + x_s^*) \tag{6.2}$$

subject to the constraints of (3.1), while $r = C / \sum_s N_s$, and $V = vC$. The regression now weighs the survey data in accordance with their estimated frequency of occurrence in the census. In the dual program (3.2), the weights are implemented by keeping the a_s and a_s^*-coefficients on the interval $[0, C N_s / \sum_s N_s]$. There is no modification in the weight of the *vC*-constraint on coefficients in (3.2). We note that the estimation will actually be as if the survey consisted of N_s replications of every observation *s*, albeit that the statistics on its reliability will consider *S* rather than $\sum_s N_s$ observations.

[14] Recall that the kernel function is taken to be integrable.

Next, the geographical aggregates that make up the districts on the map can be evaluated as:

$$Y_g = \frac{\int_{X_g} [\sum_s (a_s - a_s^*) k(x_s, x) + \sum_j b_j f_j(x)] dG(x)}{\int_{X_g} dG(x)} \tag{6.3}$$

As shown in Keyzer (2000), the same redressing weights (6.1) apply to kernel density regression, and even to conventional maximum likelihood estimation, since they only affect the empirical distribution.

Finally, we mention that the classification equations of section 4 are well suited for application in the context of poverty maps, as the census can now be used to evaluate the probability of an individual falling in a particular class h, on the basis of a model that was estimated on a smaller sample. The probability of an individual in region g falling in class h is calculated from (4.5) as:

$$P_g^h = \frac{1}{N_g} \int_{X_g} t_h(x) dG(x) \tag{6.4}$$

where $t_h(x) = \{1 \text{ if } h = arg\,max_c\, g^h(x) \text{ and } 0, \text{ otherwise}\}$. Note that discrete interventions (say, treatment/no treatment) can be dealt with as part of the classification. Furthermore, the classification can be used to select the best set of explanatory variables, and in particular to test how well income data can track shifts in classification.

6.2 Testing the match between the survey and the census variables

Note that the variables of the regression function *f(x)* may consist of a subset of those in *G(x)*. For example, the survey need not be geo-referenced. So far, we have supposed that all variables are common. This was not restrictive since variables that do not appear in the survey can be thought of as appearing with zero coefficients in the regression function and with zero effect in the kernel function. However, for the mapping to be meaningful, the distributions of common variables should match in both data sets, and for this we need to represent them separately. The kernel function can also be used to test this and to evaluate, via smoothing, the density at a point z_c of a representative sample of size $\tilde{C}$ from the census :

$$\ell_c = \frac{1}{S} \sum_s k(\tilde{x}_s, z_c), \tag{6.5}$$

and the census density:

$$h_c = \frac{1}{\tilde{C}} \sum_{c'=1}^{\tilde{C}} k(z_{c'}, z_c). \tag{6.6}$$

where $\tilde{x}_s$ denote the vector of observations of the variables for which data are available in the survey, and z_c the corresponding vector of the same variables for every of the $\tilde{C}$ observations of the census (we write $\tilde{C}$ to distinguish from the constant C in kernel learning).

Note, first, that if the kernel function is a density, h_c will lie close to unity in a representative sample, and second that values of ℓ_c that systematically lie below h_c signal misspecification, as opposed to lack of representativity. The more both distributions lack overlap, the clearer this signal. Furthermore, by construction:

$$L = \frac{1}{S} \sum_s N_s = \frac{1}{\tilde{C}} \sum_c \ell_c , \tag{6.7}$$

and a value far below unity signals a problem. To test compatibility of the survey and census distributions, the hypothesis would be that

$$h_c = \ell_c + e_c, \tag{6.8}$$

where e_c are iid and $N(0, s^2)$. We test, first whether the mean e differs significantly from zero, using an estimate of the standard deviation, and through an F-test whether the deviation between both distributions is sufficiently small:

$$F = \frac{\sum_c e_c^{\,2}}{\sum_c (h_c - \bar{h})^2}. \tag{6.9}$$

If the data set fails this test, the coherence between survey and census information is insufficient to warrant the construction of a poverty map. Clearly, the tests can be performed for individual variables, but the kernel formulation also permits to do this, consistently, for a joint kernel. For single variables, the likelihoods (6.5) and (6.6) could be depicted in one graph.[15]

However, it must be stressed at this point that there exists no technical way to distinguish between lack of representativity, that can be addressed by reweighting, and lack of definitional compatibility between census and survey variables. All one can do is point to differences in the distributions. For this one possibility is to compare the moments of both distributions. Another is to conduct a transformation of variables, say, expressing them as deviations of the means, and interpreting the effect of the transformation on the F-ratio. Finally one could allow for a single conversion factor on the survey variable, treat the F-ratio as function of this factor, and identify

[15] Here we have supposed that the kernel function is given, with a window size q determined on the basis of the census. Alternatively, one could consider evaluating (6.5) with the (larger) window size of the survey.

the factor value that gives the best ratio. If this is far away from unity, the matching would seem problematic for the variable concerned.

Section 7
Conclusion

Both under kernel-based optimization (3.2) and under kernel-based smoothing (5.1), the role of the kernel is to define a geometry for extrapolation from the empirical to the true distribution. This cannot be done on the basis of the empirical data alone. The probability bound (3.10) is even fully independent of the actual observations and can offer no more than a signal of reliability. In technical applications of kernel learning this is not a serious problem, since the availability of observations is virtually unlimited, which makes it possible to test the convergence of estimators as the data set increases. In economic applications, the same holds when function estimation is used as part of the algorithmic toolkit for solving, say, infinite horizon models by optimal control rules or by value functions in dynamic programming. By the same token, when data are scarce, the gain from applying kernel based methods would seem limited.

We also mention as a basic limitation that the true model is taken to be a fully deterministic, single valued function. This implies that all errors are because lack of data inhibits the knowledge of the true function, rather than that errors in data or in the model bias the estimate. In situations where different values of the dependent variable correspond to the same, possibly discrete, value of the independent variable, it may be necessary to reject this hypothesis.

Appendix
The dual quadratic program

Derivation of the dual quadratic program is standard. We present it in the Appendix because the book by S&S is a first edition that happens to contain numerous typos and errors, especially in the formulation of primal and dual programs. Specifically, S&S refer to the unscaled eigenfunctions in the formulation of the primal program (3.1), (e.g. p.116 and 117, whereas the form on p. 270 is correct) and the derivation of the dual program on p. 174-175 does not lead to the pair of programs (3.1)-(3.2).

We consider the primal quadratic program:

$$\begin{aligned} &\min_{q\geq 0, w} \frac{1}{2} w^T w + e^T q \\ &subject\ \ to \\ &\quad Aw + Bq + \leq d \qquad (p\ ) \end{aligned} \quad , \tag{A.1}$$

where p is a Lagrange multiplier. The problem has a Lagrangean, which is to be maximized w.r.t. w and q and minimized w.r.t. p:

$$L(w,q,p\ ) = -\frac{1}{2} w^T w - e^T q - p^T (Aw + Bq + d) \tag{A.2}$$

and has first-order conditions w.r.t. w:

$$w = -A^T p \tag{A.3}$$

and for the w-terms in the Lagrangean:

$$\begin{aligned} &\frac{1}{2} w^T w = \frac{1}{2} p^T A A^T p\ , \\ &p^T A w = -p^T A A^T p\ , \end{aligned} \tag{A.4}$$

Substituting these terms in the Lagrangean yields:

$$L(w,q,p\ ) = \frac{1}{2} p^T A A^T p - p^T d - (e^T + p^T B) q\ , \tag{A.5}$$

to which corresponds the (Wolfe) dual quadratic program:

$$\min_{p \geq 0} \frac{1}{2} p^T A A^T p - d^T p$$

subject to (A.6)

$$e + B^T p \geq 0 \qquad (q)$$

Kernel learning

We are now ready to return to the primal program (3.1), and write it in matrix form. For this, we define the $S \times r$ matrix of scaled eigenfunction values

$$Y = [y_{si}] = [y_i(x_s)],$$

and the $S \times m$ matrix of values of the parametric terms

$$F = [f_{sj}] = [f_j(x_s)].$$

Hence, we can (3.1) in matrix form as:

$$R = \min_{w;b,b^* \geq 0; e \geq 0; x, x^* \geq 0} \frac{1}{2} w^T w + k^T b + k^T b^* + Ve + ri^T x + ri^T x^*$$

subject to (A.7)

$$-Yw - Fb + Fb^* - ie - x + y \leq 0 \qquad (a)$$
$$Yw + Fb - Fb^* - ie - x^* - y \leq 0 \qquad (a^*)$$

where i is a vector of length S, with unit elements (in program (6.2) it has elements N_s). This enables us to define the corresponding vectors in (A.1), $e^T = \begin{pmatrix} k^T & k^T & V & ri^T & ri^T \end{pmatrix}$, $x^T = \begin{pmatrix} b^T & b^{*T} & e & x^T & x^{*T} \end{pmatrix}$, $d^T = \begin{pmatrix} y^T & -y^T \end{pmatrix}$, $p^T = \begin{pmatrix} a^T & a^{*T} \end{pmatrix}$, as well as the matrices

$$A = \begin{bmatrix} -Y \\ Y \end{bmatrix}, \; B = \begin{bmatrix} -F & F & -i & -I & 0 \\ F & -F & -i & 0 & -I \end{bmatrix},$$

and to obtain the dual quadratic program, as follows. The objective verifies:

$$\frac{1}{2}\begin{pmatrix} a^T & a^{*T} \end{pmatrix}\begin{bmatrix} -Y \\ Y \end{bmatrix}\begin{bmatrix} Y^T & -Y^T \end{bmatrix}\begin{pmatrix} a \\ a^* \end{pmatrix} - \begin{pmatrix} y^T & -y^T \end{pmatrix}\frac{1}{2}\begin{pmatrix} a \\ a^* \end{pmatrix} =$$

$$= \frac{1}{2}(a^T - a^{*T})YY^T(a - a^*) - y^T(a - a^*) = \frac{1}{2}(a^T - a^{*T})K(a - a^*) - y^T(a - a^*),$$

while the constraints read

$$\begin{pmatrix} k \\ k \\ V \\ ri \\ ri \end{pmatrix} + \begin{bmatrix} -F^T & F^T \\ F^T & -F^T \\ -i^T & -i^T \\ -I & 0 \\ 0 & -I \end{bmatrix}\begin{pmatrix} a \\ a^* \end{pmatrix} \geq 0,$$

implying that (A.6) becomes

$$R = max_{a,a^* \geq 0}\, y^T(a - a^*) - \frac{1}{2}(a^T - a^{*T})K(a - a^*)$$

subject to

$$\begin{aligned} F^T(a - a^*) &\geq -k && (b^*) \\ F^T(a - a^*) &\leq k && (b) \\ i^T(a + a^*) &\leq V && (e) \\ a &\leq ri \\ a^* &\leq ri \end{aligned} \qquad \text{(A.8)}$$

as required, and where the k -constraints reduce for $k = 0$ to:

$$F^T(a - a^*) = 0 \qquad (g) \qquad \text{(A.9)}$$

where $g = (b - b^*)$. In (A.8) all multipliers are bounded $k > 0, V > 0$ and $r > 0$, beecause Slater's constraint qualification is met. If $k = 0$ they are bounded provided F^T has rank *m*.

Bias and consistency

We check whether the SV-estimate possesses two properties that are generally considered essential in maximum likelihood regression. First, the mean error is not necessarily equal to zero, be it calculated as the difference between the estimated function and the observation, or as mean hard error $\frac{1}{S}\sum_s(x_s - x_s^*)$. However, it is relatively straightforward to impose such constraints. For example, to ensure a zero mean hard error, it suffices to relax the bounds in the dual program by a scalar variable h of arbitrary sign.

$$a \le (r + h)i$$
$$a^* \le (r - h)i \qquad \text{(A.10)}$$

Secondly, with respect to the consistency of prediction, we note that unlike the case in, say, ordinary least squares, where the dimension of the $X^T X$ -matrix and the $X^T y$ vector do not change as the number of observations rises, in SV-regression the parameter vector itself is of infinite dimension as S goes to infinity. In this case, we have to write the problem in infinite dimension, with the finite dimension as a special case, and consider a sequence of ever less restricted problems as S goes to infinity.

The dual program (A.8) can be interpreted as a problem with the variables replaced by $\boldsymbol{a}^S$ and $\boldsymbol{a}^{*S}$ as infinite dimensional vectors, with the restriction that the elements beyond S are zero. This shows that the series of program generates a monotonically rising sequence of nonnegative values R^S. Since this is a Cauchy sequence, it converges, proving consistency of the estimator R^S. Consistency of the estimator of $f(x)$ follows directly from the Representer Theorem. However, there is no natural concept of consistency for the estimation of the individual coefficients a_s and a_s^*.

References

Ben-Hur, A., D. Horn, H.T. Siegelmann, V. Vapnik (2001) 'Support Vector Clustering',*Journal of Machine Learning Research*, 2:125-137.

Brooke, A., D. Kendrick, A. Meeraus, and R. Raman (1998) GAMS: a user's guide. GAMS Development Corporation, 1998, Washington, USA.

Collobert, R. and S. Bengio (2001) 'SVMTorch: Support Vector Machines for large-scale regression problems', *Journal of Machine Learning*, 1:143-160.

Ermoliev, Yu.M. (1988) 'Stochastic quasigradient methods', in Yu. Ermoliev R.J.B. Wets, eds. *Numerical Techniques for stochastic optimization.* Berlin: Springer.

Ermoliev, Yu., M.A. Keyzer and V. Norkin (2002), "Estimation of econometric models by risk minimization: a stochastic quasigradient approach", IIASA Interim Report IR-02-021, IIASA, Laxenburg, Austria.

Genton, M.G. (2001) 'Classes of kernels for machine learning: a statistics perspective,' *Journal of Machine Learning Research*, 2:299-312.

Greene, W.H. (1997) *Econometric Analysis. Third Edition*. London: Prentice Hall.

Haerdle, W. (1993) *Smoothing techniques, with implementation in S.* Berlin: Springer.

Hentschel, J., J. O. Lanjouw, P. Lanjouw and J. Poggi (2000) 'Combining Census and Survey Data to Trace Spatial Dimensions of Poverty: A Case Study of Ecuador', The World Bank Economic Review, Vol. 14, No. 1: 147-165.

Judd, K. (2001) 'The parametric path method: an alternative to Fair-Taylor and L-B-J for solving perfect foresight models.' Stanford, CA: Hoover Institution.

Keyzer, M.A. (2000) 'Reweighting Survey Observations by Monte Carlo Integration on a Census,' Stichting Onderzoek Wereldvoedselvoorziening. Staff Working Paper no. 00.04, the Vrije Universiteit, Amsterdam.

Lancaster, P. and M. Tismenetsky (1985) *The theory of matrices, Second Edition.* New York: Academic Press.

Schölkopf, B. and A.J. Smola (2002) *Learning with kernels: support vector machines, regularization and beyond*. Cambridge, MA: MIT Press.

Tarozzi, A. (2001) 'Estimating Comparable Poverty Counts from Incomparable Surveys: Measuring Poverty in India', Princeton University.

The Centre for World Food Studies (Dutch acronym SOW-VU) is a research institute related to the Department of Economics and Econometrics of the Vrije Universiteit Amsterdam. It was established in 1977 and engages in quantitative analyses to support national and international policy formulation in the areas of food, agriculture and development cooperation.

SOW-VU's research is directed towards the theoretical and empirical assessment of the mechanisms which determine food production, food consumption and nutritional status. Its main activities concern the design and application of regional and national models which put special emphasis on the food and agricultural sector. An analysis of the behaviour and options of socio-economic groups, including their response to price and investment policies and to externally induced changes, can contribute to the evaluation of alternative development strategies.

SOW-VU emphasizes the need to collaborate with local researchers and policy makers and to increase their planning capacity.

SOW-VU's research record consists of a series of staff working papers (for mainly internal use), research memoranda (refereed) and research reports (refereed, prepared through team work).

Centre for World Food Studies
SOW-VU
De Boelelaan 1105
1081 HV Amsterdam
The Netherlands

Telephone (31) 20 - 44 49321
Telefax (31) 20 - 44 49325
Email pm@sow.econ.vu.nl
www http://www.sow.econ.vu.nl/