Rule-based and support vector (SV-)regression/classification algorithms for joint processing of census, map, survey and district data

by

M.A. Keyzer
## Contents

Abstract ................................................................................................................................. v

1. Introduction ...................................................................................................................... 1

2. Gridding algorithms ........................................................................................................ 5
   2.1 Aims .......................................................................................................................... 5
   2.2 Constrained scaling ................................................................................................. 6
   2.3 Priority allocation ..................................................................................................... 7
   2.4 Incorporating structural properties at grid-level ....................................................... 8

3. Regression ....................................................................................................................... 10
   3.1 Background .............................................................................................................. 10
   3.2 Short summary on SV-regression ............................................................................. 11
      3.2.1 Kernel learning ................................................................................................ 11
      3.2.2 SV-regression ................................................................................................ 13
   3.3 Accounting for district means in SV-regression ....................................................... 17
      3.3.1 Choosing between treatment of district means as data or as constraints ......... 18
      3.3.2 SV-regression, district means as constraints .................................................. 20
      3.3.3 Non-linearity in \( \beta \); treatment of latent variables ...................................... 21
   3.4 Mollifier (kernel smoothing), no processing of district means ................................ 22
   3.5 Stability of prediction: mapping of errors and bagging ........................................... 22
   3.6 Probability and likelihood bounds .......................................................................... 23
   3.7 Census weights on survey observations .................................................................. 25
   3.8 Endogeneity and Autocorrelation .......................................................................... 25

4. Classification .................................................................................................................... 29
   4.1 Short summary on classification ............................................................................. 29
      4.1.1 Rule-based classifiers ....................................................................................... 29
      4.1.2 SV-classification ............................................................................................. 29
   4.2 SV-classifiers of binary, single and multiple class choice ........................................ 31
      4.2.1 Binary choice .................................................................................................. 31
      4.2.2 Single Class choice: quantile estimation ......................................................... 35
      4.2.3 Multiple class choice: class wise versus joint estimation ............................... 35
      4.2.4 Multiple class choice: multistage form ........................................................... 37
   4.3 Classification by kernel smoothing (mollifier) ........................................................... 38
   4.4 Nearest neighbor ....................................................................................................... 39
   4.5 District data in SV-classifier .................................................................................... 39
   4.6 Ranking while accounting for district data ............................................................... 41
   4.8 Probability and likelihood bounds .......................................................................... 42

5. Micro-simulation and prediction of district aggregates ................................................... 44

Appendix: GRCP, a GAMS-controlled package for Gridding, Regression, Classification and Plotting ............................................................................................................. 46
   A. Gridding and plotting of maps ................................................................................... 46
Abstract

The paper considers the updating datasets with census data and grid maps, as well as the parameter estimation and micro-simulation of models based on these datasets, while accounting jointly for (i) survey data such as expenditures from households, rainfall data from weather stations and data on land classification; and (ii) aggregate data at district or sector level as published by statistical agencies. Various rule-based gridding, regression and classification algorithms to conduct these tasks are presented that rely on kernel smoothing, nearest neighbor and support vector techniques. Beyond this incorporation of census and grid map data within the survey based regression/classification itself, as opposed to their use in prediction only, the intended contribution is to allow for formulations with discrete data other than single, binary and multiple class choice, such as ranking with known and unknown class bounds and a consumer model with discrete choice and to present GRCP, a software package designed for this purpose, through which the user can readily navigate between gridding, regression and classification, controlling by simple GAMS commands all operations from the basic data down to the automatic preparation of tables and geographic maps with appropriate legends and 2D and 3D plotting of functions.
1. Introduction

Applied economists engaged in statistical interpolation exercises to compile consistent data sets at regional or national level, or in the modeling of individual behavior, often have at their disposal data from four distinct sources: (i) a (usually somewhat outdated) a geographical map; (ii) a (similarly outdated) a household census; (ii) more recent data at macro (village, county, province etc., or sectoral)-level; (iii) one or more dedicated household surveys of varying vintage.

While recognizing the merits of adopting an integrated approach for the use of data from such diverse sources, academic researchers tend to view this as an art beyond the domain of formal analysis, and to delegate the task to their junior staff. Indeed, the account in scientific publications and policy reports tends to start with an already “compiled” data base and usually focuses on a single data type: the household data set; the geographical maps; the district aggregates. Econometric studies generally focus on estimation and prediction at either micro (survey)-level or macro (district or sector) level, hence avoiding difficulties associated to the heterogeneity of data sources. Micro-simulation on micro-data is most directly concerned with the relationship between micro and macro since it conducts calculations at micro-level, often a detailed survey, to obtain through summation estimates or predictions at more aggregate level. Micro-simulation often relies on rule-based assignments through an a priori given function that links the variables of interest to underlying census data, but some authors use for this a regression function previously estimated from a survey.

Highly prevalent in data base compilation are the various interpolation exercises to fill data gaps, which are in essence rather similar to micro-simulation based predictions, as they conduct calculations at values of exogenous variables for which no observation of the endogenous variable is available, e.g. for households in the census or grid points on the map not represented in the survey data set. Here also rule-based approaches are most common but regressions gradually become more prominent. For example, Elbers et al. (2003) and Deaton (1997) have, in poverty mapping and national accounting, respectively, applied regression-based micro-simulation to obtain estimates of district averages.

These micro-approaches produce their own macro-estimates through aggregation. While one might maintain the viewpoint that there hardly is such a thing as a macro-observation, since all aggregates should be obtained from summation over micro-data, the practice is that statistical agencies through a variety of techniques update their macro-data far more often than they release new censuses, and clearly policies have to be formulated also for years in which no new census is conducted.

The challenge is now to make use of these sources, micro as well as macro, in a consistent way at the successive stages of database construction, model estimation, and prediction. We note that these stages have in common that each of them actually amounts to finding best, say, risk minimizing values for some unobserved quantities on the basis of observations available from various sources.

The procedures are primarily designed for two fields of application. One is household modeling. This is the classic domain of micro-simulation. The questionnaires of household censuses usually include a question on the district of residence that establishes a relatively crude linkage from households to spatial location for which data may have been compiled from a different source. They may also include questions relating to the main sector of activity that can establish a linkage to sector data. In some cases, no macro-data are available at district level. The algorithms to be
described apply nonetheless, drawing on what is available at macro-level for calibration purposes and using the spatial address from the census for cartography only.

Spatially explicit modeling is the other intended field of application. Many issues in spheres as diverse as economic development, transport, hydrology, environment and health have an important spatial dimension affected as they are by distances, topography, climate, soil conditions and so on. Whereas multiregional modeling subdivides a territory in polygons and describes what happens in each of these, spatially explicit conducts a much finer subdivision into (square) grid cells (or pixel) and need interpolated data or predictions at this level.

The present paper is concerned with the specification of numerical procedures that can manage the sequence of stages in an integrated way, starting from the original data down to the final tables and maps, accounting simultaneously for both micro-data from smaller but more detailed surveys (from say, households or weather stations), and aggregate data (say, by district or by economic sector).

The questions underlying these data sets can be of various kinds that critically define the statistical approach needed to process the answers.

(1) For some questions, such as say, “what were your expenditures on fish products last months?” the answer will be real-valued unrestricted. Explaining such an answer ($y$) from other answers ($x$) gives rise to regression problems, i.e. to the estimation of the function $y(x)$.

(2) The answer may be real-valued but with limited range (e.g. on the 0-1 interval in case we ask for the share of fish expenditures in total). This leads to a regression problem with a limited dependent variable.

(3) The answer may be of the yes-or-no type. This leads to a binary classification problem.

(4) The answer may be of the multiple class type (e.g. single/married/divorced/widow), leading to a multiple classification problem.

(5) The answer may be of a qualitative ranking nature (e.g. 1,2,3,...prize): a ranking problem.

(6) The answer may point to quantitative brackets, e.g. expenditure categories: e.g. $1 = [0-100 \text{ US dollar}), 2 = [100,200), 3 = [300 and higher}$, leading to a ranking problem with given class boundaries.

While each type of answer needs a different technique, this leaves ample room for actual choice of technique. Furthermore, as various nomenclatures co-exist and overlap, we briefly some key subdivisions. One is the distinction between rule-based decision rules (gridding procedures/expert rules) that do not require survey data and estimation techniques (regression/classification) that do, as partly competing, partly complementary approaches. Gridding and regression techniques apply to the case where observations on the variable of interest (the dependent variable) are real-valued (case (1-2) above); expert rules and classification techniques to all other cases.

Gridding, also known as downscaling is popular in the domain of GIS. It uses an a priori given algorithmic procedure to distribute real-valued district data, say, on total production of a commodity over the households in a population census or over the grid cells of a map.
In situations where, in addition, real-valued data on the (dependent) variable available at district level are also (or only) available from a survey, i.e. a subset of the individual households or grid cells, regression can be applied. Obviously, results from regression might be unsatisfactory, which may motivate recourse to rule-based gridding eventually.

Classification is needed when the dependent variable falls in category (3)-(6). In this case, observations only indicate to which class the dependent variable belongs. Rule-based decision now amounts to invoking given classification rules, whereas estimation prior to prediction adjusts the coefficients of these rules so as to fit survey data.

The gridding techniques to be considered add to the common proportionate distribution rules, options of priority allocation and bounded distribution. As regression and classification techniques we distinguish three approaches: (i) Support vector (SV) regression and classification (e.g. Christiani and Shawe-Taylor, 2000; Schoelkopf and Smola, 2002), under two different options depending on whether eigenfunctions or kernel functions are used; (ii) OLS regression of a polynomial form (only for regression); and (iii) Mollifier or Nadaraya-Watson kernel smoothing regression and classification (Haerdle, 1993); that are parametric, semiparametric and nonparametric, respectively. Furthermore, for classification we also allow for the nearest neighbor technique (Devroye et al. 1996).

SV-regression has the advantage that it can fit flexibly, is less sensitive to outliers and explicitly accounts for the similarity of neighbors (spatially or otherwise) than maximum likelihood methods. The OLS-option is introduced for easy comparison with SV-regression and to deal with situations where the number of survey observations is extremely large; the mollifier is very useful at the stage of data exploration before any function structure is imposed. Nearest neighbor classification is easy to implement and has remarkable properties.

The contribution of this paper are first, the incorporation of census and grid map data within the regression/classification itself, as opposed to their use in prediction only (e.g. Hentschel et al., 2000 and Tarozzi, 2001). Second, we incorporate within the SV-setting formulations with discrete data other than single, binary and multiple class choice, such as ranking with known and unknown class bounds and a consumer model with discrete choice. Finally, we present dedicated software developed for this purpose.

The interpretation of satellite images is one example. While the frequency spectrum of the image at every pixel generates a map (population census) for the independent variables in the regression, at points where ground observations of the dependent variable are available (e.g. the grain content of a crop on the field for regression, soil type for classification), pairing with the independent variables defines a training set. The task is to estimate the parameters of the regression/classification function using both this training data set and the macro-data. Next, the prediction task is to fill through interpolation all the points between those of the training set. Alternatively, the regression/classification refers to a behavioral model, and applied in micro-simulation.

With this characterization, the estimation (statistical learning) approach might seem superior but in our view both approaches are complementary, because unless the training set can be made very large, which as discussed above is rarely the case, estimation needs well-articulated rules to start from and can only vary some of the coefficients of these rules.
The paper proceeds in the order that most naturally suits the techniques applied. In section 2, we present various gridding procedures; in section 3, we turn to regression techniques that allow for inclusion of survey data in addition to district and census and geographical information. Besides straight least squares, we consider distribution-free methods risk minimizing techniques from kernel learning and averaging techniques (mollifier) from kernel smoothing. In Section 4, we turn to classification, starting from expert-based rules and the nearest neighbor classification algorithm, a widely used distribution free technique with remarkable convergence properties, that like the mollifier requires survey data but needs no parameter estimation prior to prediction, before turning to various estimation techniques that parallel those of Section 3 for various data configurations. Finally, Section 5 briefly considers micro-simulation and the prediction of district aggregates for these configurations.

The appendix contains a user guide to GRCP, the new software developed in this project to accommodate these data configurations, in which a short GAMS-program (one for rule-based and one for regression) controls all operations, from the basic data down to the automatic preparation of tables and geographic maps with appropriate legends and the 2D and 3D-plotting of functions. The actual operations are conducted in Fortran, SAS and GAMS, depending on the task, without any user involvement. In the pattern recognition literature much attention is currently devoted to the design of recursive algorithms for very large-scale applications. We do not discuss this aspect because the applications envisaged here do not require on-line solution and can, therefore, be accommodated by conducting preparatory calculations in Fortran that build matrices whose dimension does not exceed the total number of survey and district observations and is independent of the size of the census and grid map.
2. **Gridding algorithms**

2.1 **Aims**

We consider a grid \( G \) of sites indexed \( s, s = 1, ..., M \), constituting a region \( G = \{1, ..., M\} \), partitioned into non-overlapping districts \( D_s \): \( \bigcup_s D_s = G, D_s \cap D_{s'} = \emptyset \) for any \( s \neq s' \), indexed \( d \), while \( d_s \) denotes the district to which \( s \) belongs.\(^1\) We assume that grid-level (census) data \( (x_{1s}, ..., x_{Hs}) \) are given. The micro-model we are concerned with reads:

\[
y_s = f(x_s, \eta)
\]  

(2.1)

with parameters \( \gamma \) to be estimated and to obtain from this function predictions \( y_s \) on the full region \( G \), preferably jointly with error bounds on these predictions. There are no observations on \( y_s \). A rule-based approach assumes a full specification of the function \( f \) in which \( \gamma \) has district wise elements that adjust to meet the following aims:

(a) *imposing district totals.* Since data are generally collected at district-level, there is a regular need to update lower level data with new information, so as to ensure that the grid-values sum to the known totals. Often, the policy analyst is only interested in these totals, and the deeper grid merely serves for the parameterization of the model under consideration, which is kept spatially explicit nonetheless to maintain the structural properties induced by the spatial dimension, and the inherent prediction errors induced by informed guesses at the level of the grid are taken to vanish through aggregation. Hence, under rule-based scaling, the imposition of district totals amounts to assuming that the district indicator \( \delta^d_s = 1 \) if \( d_s = d \) and 0 otherwise, is an element of the list of determinants \( x_{hs} \) in (2.1), with district wise calibration parameters \( \lambda^d \) associated to it. This leads to the district wise identities:

\[
y^d = \sum_s \delta^d_s f(x_s, \eta) = \sum_s \delta^d_s \hat{y}_s(\lambda^d),
\]  

(2.2)

where \( \hat{y}_s(\lambda^d) \), represents the rule for adjusting the individual value, for given values of \( x_s \), and an algorithm is specified to solve for \( \lambda^d \). This function can have various functional forms, including the linear one. Monotonicity w.r.t. \( \lambda^d \) is the major requirement to ensure uniqueness of the solution.

(b) *a priori bounds at grid-level.* Bounds can be incorporated in the specification of \( \hat{y}_s(\lambda^d) \) through a truncation that keeps the values within an acceptable range. Clearly, the algorithms have to deal with the resulting non-linearities. The class to be discussed below has the advantage that it can, under suitable linearity assumptions, find a unique exact solution in a finite number of iterations and remains efficient when the number of sites is very large.

---

1 In a population census, the full population corresponds to the “region”, and is partitioned in various ways into aggregates, by age, sex, education etc. (the “districts” or “sectors”).
(c) priorities among grid cells. Algorithms based on (2.2), to be discussed further in section 2.2, tend to keep the distribution of the already gridded variable (the reference variable) as much unchanged as possible. By contrast, we may consider a procedure that allocates (part of the variable) more than proportionately to site with a higher “payoff”, within given bounds. This is considered in section 2.3.

(d) crosslook. The grid map of one variable should remain compatible with that of others. Hence, the bounds on one variable could follow from grid maps for other variables. For example, the population map dictates that no economic activity can take place at unpopulated sites, and urban infrastructure is unlikely to be located at sites with low population density. This can be accounted through (e.g. population-based) thresholds. More importantly, in case no a priori distribution is available for a variable, it will be necessary to borrow from the distribution of another variable. Hence, the need to consider all gridding in a single sequence (see Appendix A).

(e) feedback from grid to district. The adjustment procedure should not be such that district-level information always takes precedence over the grid-level. It should accommodate a confrontation of both, and, therefore, possess a facility to aggregate grid-level back to district-level.

(f) comprehensive script. Since the number of grid cells in every district can be very large trial-and-error procedures should be avoided. The software should work with a comprehensive script controlling the full application, and be user-friendly with a limited number of commands, so as to help the modeler keep an overview of the full process from basic grid-files and district totals until the final maps and model files. This eases the integrated implementation of multidisciplinary projects as well as documentation and makes it simple to reproduce results of a sometimes long and complex chain of calculations. The operation of such software is described in Appendix A.

Below we consider two classes of rules for distributing district totals over a spatial grid, to be invoked within the script: constrained scaling and priority allocation. While constrained scaling maintains to the extent possible a given distribution, priority allocation maintains full flexibility and assigns, within bounds, the district totals to the sites with highest payoff.

2.2 Constrained scaling

The constrained scaling rule amounts to shifting the given distribution of the reference variable, insofar as this variable exceeds a given threshold, truncating it so as to satisfy given bounds, while pursuing the shift until the total value attributed to cells in each district equals the given district total.

The gridding algorithm finds a pair \((y, \lambda)\) that solves:

\[
y_s = \min(\max(a_s + b_s \lambda^{d_s}, y_s), \bar{y}_s),
\]

\[
\sum_s \delta^d_s y_s = Y^d,
\]

where \(Y^d\) is the given district total of the variable concerned; \(\delta^d_s = 1\) if \(d_s = d\) and 0 otherwise; \(b_s = a_s\) in case of proportionate shift of the distribution, and \(b_s = \mu_s^{d_s}\) (the mean:
\[ \mu^d = \frac{\sum_s \delta^d_s a_s}{\sum_s \delta^d_s} \] in case of additive shift, and \( a_s \) is the scaled value of the elements of the reference variable \( x_s \) that exceed the threshold \( x_s \):

\[ a_s = \rho^d \hat{x}_s \]

where \( \hat{x}_s = x_s \) if \( x_s \geq x_s \) and \( 0 \) otherwise,

and

\[ \rho^d = \frac{y^d}{\sum_s \delta^d_s \hat{x}_s} \].

It is assumed that \( y_s = 0 \) whenever \( \hat{x}_s = 0 \). The algorithm proceeds by applying a variant of the Newton-Raphson routine that in view of the linearity of the function inside the min-max, has finite termination. For the sites at which the min-max operators are not effective, denoted by a unit\(^2\) value for \( \hat{\delta}^d_s \) (with \( 0 \) otherwise), we can calculate a target value for \( \hat{\lambda}^d = \frac{y^d - \sum_s \hat{\delta}^d_s a_s}{\sum_s \hat{\delta}^d_s b_s} \) and use it for the next iteration, provided it falls within the best-so-far (lowest deficit and lowest surplus) interval on \( \lambda_d \), otherwise, bisect, i.e. use middle of interval as new value.

### 2.3 Priority allocation

The constrained scaling rule maintains the given distribution to the extent possible. Indeed, problem (1.1) can be written as a maximization problem subject to constraints, which illustrates that the procedure keeps the proportions

\[ \min_{\mu_s, v_s, y_s \geq 0, \lambda^d_s} \sum_s (\mu_s + v_s) \]

subject to

\[ y_s = a_s + b_s \lambda^d_s + \mu_s - v_s \]

\[ y_s \leq y_s \leq y_s \]

\[ \sum_s \delta^d_s y_s = y^d \].

The priority allocation assigns the district-total to the sites with the largest payoffs \( \hat{x}_s \), until their upper bound.

\[ \max y_s \sum_s \hat{x}_s y_s \]

---

\(^2\) In applications, especially when the micro-data originate from a household survey, grid variable will often be on a per capita, per household or per hectare basis. This can be accommodated by redefining the weight \( \delta^d_s \) in the district-constraint as measuring the number of people, households, and hectares, respectively, instead of being a 0-1 coefficient.
subject to

\[
\begin{align*}
\underline{y}_s & \leq y_s \leq \bar{y}_s \\
\sum_s \delta^d_s y_s &= y^d .
\end{align*}
\]

Here it is assumed that \( \underline{y}_s = 0 \) whenever \( \hat{x}_s = 0 \). The algorithm solves for surpluses \( z_s \) above the lower bounds, i.e. addresses the equivalent problem:

\[
\begin{align*}
\max_{z_s \geq 0} & \sum_s \hat{x}_s z_s \\
\text{subject to} & \sum_s \delta^d_s z_s = y^d - \sum_s \delta^d_s \bar{y}_s ,
\end{align*}
\]

by ranking the payoffs in decreasing order, and visiting sites in that order, assigning a maximal value \( (\bar{y}_s - \underline{y}_s) \) until the site where the cumulated total reaches the given value \( (y^d - \sum_s \delta^d_s \bar{y}_s) \). That site receives the balance and the remaining sites nothing.

### 2.4 Incorporating structural properties at grid-level

We stress that gridding procedures cannot be applied blindly as they must respect some basic properties of the spatial structure under consideration, which must be reflected as restrictions within the procedure and were not incorporated in the treatment so far. These restrictions fall in two classes.

The first class relates to on-site restrictions. A straightforward example is the spatial property of the grid itself that it is a geographic entity of a given surface. All adjustments of variables expressed as a surface need to account for this. The second relates to between-site restrictions, usually between neighboring sites. For example, spatial flows must obey laws of conservation (no manna from heaven or sudden disappearance) and gravity (no flows uphill). The solutions to deal with the various configurations strongly depend on the specifics of the problem, and all sorts of dedicated gridding rules could be designed. Here we only review a few that operate on the ones specified above.

**On-site restrictions: filling layer-by-layer.** Rather than allocating the district total in one round, the total can be gridded in a sequential process. Suppose that land use on a site comprises housing, horticulture, field crops and shrub land, of which the initial areas at some site are 1, 10, 100, 200, respectively. A first step is to check that the total area given at district level agrees with the total area of the grid points. The gridding is then to adjust the land use map. Typically, the area for housing would be assigned on the basis of a given rural population map, and maintained fixed after that, to meet some district total. This would modify the area to, say, 1.05. This increase would have to be subtracted from other land uses at that site. For this one might subsequently grid, say, 90 percent of the area of horticulture, say, in proportion to the initial surfaces allocated, with, say, 90 per cent the total remaining area as site specific upper bound. This would leave some part of the allocation. After computing the remaining area free for allocation, one would start a second layer, to distribute the remaining horticultural land with a bound, and after that part of the field crops, also with a bound. And so on, until all land is distributed. This layer-by-layer
filling can be modified in several ways. The major limitation is, obviously, that the end-result is somewhat arbitrary, definitely more so than regression based allocations, as for all rule-based procedures but it has the merit of seeking to do justice to the site-specific information to the extent possible, when reconciling it with the district-totals, and is easier to control than the mechanistic iterative procedures such as RAS or entropy-methods for ensuring that a matrix with given column-sums (district totals) and row-sums (the surface of a grid cell), and initially given land allocations eventually converges to an allocation that meets these totals.

**Between-sites restrictions:** Some of the restrictions on allocations among sites can be imposed directly as bounds by site, with the relative values of these bounds reflecting the restriction. Furthermore, the allocation procedures (2.3) and (2.6) themselves obviously imply relationships between sites: in (2.3) in terms of distribution of the reference variables and in (2.6) in terms of the payoffs. However, this does not account for empirical relationships to be discussed in section 3 below, or for cumulative phenomena, such as hydrological flows. With respect to the latter, we merely observe that their treatment falls beyond the scope of the present paper and calls for the use of a spatially explicit hydrological model that visits the sites in order of decreasing altitude, within a calibration procedure, whereby model coefficients such as the retention factors are adjusted until the flows at specific points (hydrographs) meet the observations.

**Relation to RAS-scaling.** So far, no reference was made to the well known RAS-approach for distribution of row- and column totals over the cells in a matrix, which might seem a good candidate algorithm for rule-based gridding. This deserves some comment. Suppose that we are faced with a very large census that can only be processed by means of a representative sample or, that the census available needs updating. For ease of exposition we suppose also that the census only records the age and the education level of respondents. We are looking for weights on individual observations to ensure that the census sums by class (e.g. age and education) to known recently updated population totals. Let \( a \) denote the age and \( e \) the education bracket of the respondent, while \( \delta_{ae} \) describes the sample, and is 1 if the bracket applies and 0 otherwise. At the level of the population the totals \( A_a \) and \( E_e \) measure the number of individuals of age \( a \) and education \( e \), respectively. To scale the number of individuals \( \delta_{ae} \), so as to meet the restrictions, the RAS-approach of sequential proportionate adjustment would proceed as:

\[
\sum_a \lambda_{a,t-1} \lambda_{e,t} \delta_{ae} = E_e, \quad t = 1, 2, \ldots
\]
\[
\sum_e \lambda_{a,t} \lambda_{e,t-1} \delta_{ae} = A_a,
\]

given \( \lambda_{a,0} = \lambda_{e,0} = 1 \). This procedure has major drawbacks. It is iterative and hence numerically costly when applied to a large sample; moreover, for certain data configurations it may not converge; finally, the form does not allow for maintaining bounds on individual outcomes \( y_{ae} = \lambda_a \lambda_e \delta_{ae} \), as upward scaling with respect to age may coincide with downward scaling with respect to education. Alternatively, rule-based scaling would operate on the total \( B_{ae} \), possibly obtained through a RAS-procedure in case only the totals \( A_a \) and \( E_e \) are available, and apply a single, and hence far more transparent, adjustment to scale the sample, with or without bounds imposed. In short, the RAS should only be applied at the aggregate level where row and column totals are available as data, not for gridding.
3. Regression

3.1 Background

We now turn to the regression approaches, which we also refer to as statistical learning to emphasize that attention extends beyond common least square-type or maximum likelihood fitting and in fact focuses on support-vector techniques. In applications, the common setting will be that a household survey which is not geo-referenced is to be relied upon jointly with a population census that only contains geographical reference at district level (as opposed to the spatial co-ordinates of the residence), while macro-data are available for variables at both district and other levels that can link to the census. Hence, we must consider the estimation and prediction over the census, for given micro data as well as macro data that can be of any type. However, for clarity of exposition we again relate to the application in a spatial context, referring to micro-units as grid cells and to macro-units as districts.

Formally, we consider a data set consisting of a single dependent variable \( y \) and \( H \) independent (reference) variables \( x_h, \ h = 1, \ldots, H \). If grid-level data \((y_s, x_{1s}, \ldots, x_{Hs})\) are available on a sufficiently large training set \( G^o \), while data on \( x_{hs} \) are available throughout the region \( G \), it is natural to postulate as in (2.1) a predictor function:

\[
\hat{y}_s = f(x_s, \eta), \tag{3.1}
\]

with parameters \( \eta \) to be estimated and to obtain from using this function predictions \( \hat{y}_s \) on the full region \( G \) with \( M \) grid points, preferably jointly with error bounds on these predictions. In this function, the variables \( y \) and \( x \) are generally expressed per unit: per household, per capita, per hectare, etc. depending on the application, but not always, as a price could for instance enter the per capita consumption function of an individual.

The training set is not necessarily geo-referenced, but is taken to refer to data at grid level. While virtually any econometric regression technique could be applied, a distribution-free technique might be preferable, since the data-generating process is bound to possess substantial spatial correlation that cannot be characterized \textit{a priori}, and normality assumptions based on the central-limit theorem may not always be plausible at the level of the individual grid as the sample from objects within the grid may become small. Moreover, a method that allows for a flexible adaptation of the functional form would be preferred, since the availability of independent grid-variables \( x_h \) is limited, and cannot be expected to relate to the dependent variable according to some clear functional relationship derived from theory with few parameters.

We use micro-equations at individual (e.g. grid cell) level to generate matrices for regression with survey data as well as district means for dependent variables. The micro-equations may be non-linear in variables but they are linear in parameters. Furthermore, it is assumed that data (a census) are available on the joint distribution of the independent variables of the survey data within the overall population (in a statistical sense). The census is used to generate the terms of the district level functions by summation of individual functions, possibly with a weight, to be confronted with district level (macro) data. Estimation proceeds simultaneously for the micro and the macro-data. Technically, the difficulties in implementing the approach relate to:
(1) **Size.** The census data set is far too large to be processed in a regular estimation package, especially for SV-regression, since it operate on kernel matrices of dimension $M \times M$, where $M$ is the number of elements in the census. This difficulty is addressed by conducting preparatory steps (in Fortran) that generate matrices whose dimension does not exceed $(S + D) \times (S + D)$, where $S$ and $D$ denote the number survey and district observations, respectively.

(2) **Data communication.** Regression packages do not readily transfer the information (legends etc.) to plotting software to produce maps. Our facility takes care of this difficulty: once basic grid files with latitude and longitude of the points $s$ are made available in a particular SAS-readable form, jointly with polygon map with the contours of the districts, it produces maps of any variable, including an automatic formatting to a pre-specified number of classes (colors), and the associated legends with text as specified in GAMS.

(3) **Reliability.** Statistics must be provided of the probability of say, a ten percent error in prediction, so as to assess the quality of prediction at points of the map other than those in the survey data set.

Further details on the operation are given in Appendix B.

### 3.2 Short summary on SV-regression

#### 3.2.1 Kernel learning

Kernel learning, (e.g. Christiani and Shawe-Taylor, 2000; Schoelkopf and Smola, 2002) offers a well established set of tools that are well suited to accommodate parameter-free and functionally flexible regression, and to derive “Huber-robust”-estimators by neglecting $\varepsilon$-small deviations (Vapnik, 1998).

Kernel learning postulates that the relationship between an observed vector variable $x$ and an observed scalar variable $y$ obeys a constant conditional density $g(y \mid x)$. That $x$ does not give rise to a single $y$-value may be due to inherent stochasticity of the process or to the fact that $y$ is in addition to the observed variable $x$, caused by unobserved variables $z$ obeying a fixed density (as opposed to a time dependent one); $x$ is referred to as independent variable and $y$ as dependent variable.

Nonparametric kernel learning considers the estimation of the median $\mu(x)$ of this conditional density by means of a function $f(x)$. Semiparametric kernel learning treats this function as the sum of a non-parametric part $\psi(x)$ and a parametric part $\phi(x)$, both linear in to be estimated coefficients. The ($x$-dependent) median estimated is that of $q = y - \phi(x)$ for an unknown conditional density $\tilde{g}(q \mid x)$. Hence, the non-parametric term refers to the a priori (i.e. parametrically) explained part $q$ of the variation in $y$. Minimization of mean squared deviations corresponds to estimation of the conditional mean, minimization of mean absolute values to the median of the conditional density.

By contrast, the least square methods of econometrics, to the extent that they treat observations as uncorrelated, postulate in addition to the function $\phi(x)$ that $\tilde{g}$ does not depend on $x$, while the maximum likelihood methods further postulate some a priori form for $\tilde{g}$. Versions that allow for
correlation among observations essentially take linear transformation of the deviations to be independent.

The parametric and semiparametric forms of kernel learning to be considered in this paper minimize the mean absolute value of the errors in regression, to the extent that these errors exceed a given threshold \( \epsilon \), a scalar, plus some regularization term that keeps the (potentially infinite) number of parameters bounded and is to be eliminated eventually as the number of observations goes to infinity.

Here we follow Norkin and Keyzer (2005), which itself builds on Györfi et al., 2002 and Vapnik, 1998. Specifically, for \( S \) observations \( z = (x, y) \), iid sampled from some unknown but supposedly stationary joint distribution \( L(dx, dy) \) with compact support \( X \times Y \), we define, for a function \( f : X \rightarrow Y \subset \mathbb{R}^d \) from some class \( F_S \) of functionals that could depend on the number of observations. Also, let \( R(f) = E_{x,y}c(z, f(x)) \) denote a (true) risk functional. Hence, under nonparametric estimation the (true) risk minimization problem amounts to finding:

\[
f^0 = \arg \inf_{f \in F_S} R(f).
\]

As mentioned earlier, for the quadratic loss functional, \( c(z, f(x)) = (y - f(x))^2 \), as used in least squares type of estimation, it is well known that if the distribution \( L(dx, dv) \) has a conditional mean \( \mu(x) \) under fixed \( x \) and that \( \mu(x) \in F_{\mu} \) implies \( f_{opt} = \mu \) (see Györfi et al., 2002). For an absolute value loss function \( c(z, f(x)) = |y - f(x)| \) the question arises as to what the unconstrained minimizer in (3.2) of \( R(f) \) over all measurable functions \( f(x) \) amounts to. For the non-parametric form, it is the median of the distribution \( L(dx, dy) \), under the following assumption (for the semi-parametric form the assumption refers to \( q \) instead of \( y \)):

**Assumption 1** (Identifiability of the model):

(a) Assume that \( L(dx, dy) \) has conditional density \( \ell_x(y) \), and that the conditional distribution function \( L_x(t) = \int_{-\infty}^t \ell_x(y)dy \) is strictly monotonic in \( t \) and continuous on \( X \).
(b) The median \( \mu(x) \) of \( L_x(t) \) exists and is continuous on \( X \).
(c) The density \( \ell(x) = \int \ell(x, y)dy \) is non-degenerate, i.e. \( \ell(x) > 0 \) for all \( x \in X \).
(d) On the set \( X \), the expected error or disturbance function \( A(x) = E_y |y - f(x)| \) is continuous in \( x \) wherever \( f(x) \) is continuous.

Here requirement (a) says that \( y \) should have sufficient variation at every \( x \)-value where the function has to be identified; (b) limits attention to distributions with bounded conditional medians, and to true functions that are continuous; the continuity-requirement is restrictive as it for example rules out multimodal densities; and (c) rules out discrete-valued components of the vector \( x \). Hence, discontinuities cannot be accounted for; (d) as in (b) rules out multimodal kind of densities around the function \( f \). Only requirements (b) and (d) are specific to the absolute value loss function used in kernel learning.

---

\(^3\) We use \( L \) for distribution and \( \ell \) for density instead of the more common \( P \) and \( p \) to avoid confusion with prices \( p \).
The next step is to present an explicit form for the function $f$. The key element in kernel learning is that this is done in a flexible way, distinguishing a non-parametric term with an in principle infinite linear expansion of terms, a weighted sum $\sum_i w_i \psi_i(x_s)$ of the possibly unknown eigenfunctions $\psi_i(x_s)$, indexed $i$, incremented by a finite number of terms $\sum_j \phi_j(x_s)(\beta_j - \beta^*_j)$ of known functions $\phi_j(x_s)$, the parametric part of the specification, for nonnegative $\beta_j$ and $\beta^*_j$. Hence, the function is represented as:

$$f^*_S(x_s) = \sum_i w_i \psi_i(x_s) + \sum_j \phi_j(x_s)(\beta_j - \beta^*_j),$$

(3.3a)

where in (3.3a) we write $f^*_S(x_s)$ to denote that the function through its coefficients $w$ and $\beta$ depends on the sample of size $S$ and the assumed regularization factor $\lambda$ to be introduced below. The parametric part is used to represent a postulated theoretical relationship; the non-parametric part is the estimated median of the gap between $y$ and this relationship. While the $\beta$-coefficients can be constrained rather easily so as to limit the range of the parametric term, the non-parametric term has to remain unrestricted to maintain the flexibility of the adjustment. Yet, through monotonic mappings on $y$, such as the logarithm or the inverse logistic function it is possible to control the range.

The survey information on dependent variables enters via the constraints:

$$y_s \leq f^*_S(x_s) + \xi_s + \epsilon$$
$$y_s \geq f^*_S(x_s) - \xi^*_s - \epsilon,$$  \hspace{0.5cm} s \in G^0. \hspace{1cm} (3.3b)\hspace{1cm} (3.3c)

For ease of exposition, we initially suppose that the eigenfunctions are known and finite in number. In this case, the constraints (3.3) are essentially the restrictions to a linear regression problem. Indeed, if the number of eigenfunctions is much less than the number of observations, the flexibility of the function will be restricted but for $\epsilon = 0$, (3.3) would naturally fit in an OLS-type framework minimizing $\sum_{s \in G^0} g_s(\xi_s - \xi^*_s)^2$, and, therefore, also in a maximum likelihood setting.

### 3.2.2 SV-regression

To represent the actual estimation, we rewrite this in matrix terms, defining the $S \times I$ matrix of eigenfunction values ($S$ is the number of observations in the set $G^0$, $I$ the number of eigenfunctions)

$$\Psi = [\psi_{si}] = [\psi_i(x_s)],$$

(3.4a)

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

---

4 Maintaining concavity or monotonicity of the parametric term may require imposing constants on the coefficients $\beta$. Furthermore, technologically specified expert functions can be incorporated via a single term without any $\beta$-coefficient. This should be controlled directly in the GAMS-estimation program RGvBag.gms.

5 See Appendix B for details.
\[ \Phi = [\phi_j] = [\phi_j(x_s)], \quad (3.4b) \]

which in case of OLS estimation and if the rank of \([\Psi', \Phi']\) was equal to \(I + J\) would yield the estimators:

\[
\begin{pmatrix}
\hat{w} \\
\hat{\beta}
\end{pmatrix}
= \left(\left[\Psi', \Phi\right]^T \left[\Psi', \Phi\right]\right)^{-1} \left[\Psi', \Phi\right]^T y. \quad (3.5)
\]

However, for flexible forms, the number of parameters will in general exceed the number of observations, and consequently, the rank condition cannot be met. In that situation, a more general formulation must be opted for that can deal with singularity. The inequalities in (3.3) are helpful in this respect as they make it possible to ensure that the constraint set has a strict interior and, consequently, the optimization remains well behaved also when the rank becomes deficient. Common techniques, such as least squares can accommodate this but kernel learning offers the additional advantage that it explicitly addresses the representation of an unknown function by a closed form that is linear in parameters. To formulate the kernel-learning program associated with (3.4), it only remains to specify a risk criterion.

Unless indicated otherwise all errors are taken to weighted equally, i.e. with weight \(g_s = 1\) (not to be confused with the \(g\) of the density). For small samples, it may be effective to apply non-unitary weight measuring the probability of occurrence of a particular observation. In section 3.7, we consider such a weight. However, for larger samples the convergence of estimators requires these weights quickly to move to unity.

Then, the empirical risk criterion of kernel learning is written for given regularization factor \(\lambda\):

\[
R_S^2 = \frac{1}{2} \lambda \sum_i w_i^2 + \sum_j \tau_j (\beta_j + \beta_j^*) + v_\epsilon + \frac{1}{S} \sum_{s \in G^s} g_s (\xi_s + \xi_s^*) \quad (3.6)
\]

where \(\tau, v\) and the regularization factor \(\lambda\) are given positive constants (\(\tau_j = 0\) is admissible), while \(w, \beta_j, \beta_j^*, \epsilon, \xi_s\) and \(\xi_s^*\) are choice variables. Kernel learning minimizes \(R_S^2\) in (3.6), after division of all objective coefficients by \(\lambda\), subject to (3.3b,c). Hence, rewriting (3.3)-(3.4), (3.6) in matrix form it solves:

\[
R_S = \min_{w: \beta, \beta^* \geq 0; \epsilon \geq 0; \xi, \xi^* \geq 0} \frac{1}{2} w^T \Phi + \frac{1}{\lambda} \tau^T \beta + \frac{1}{\lambda} \tau^T \beta^* + v_\epsilon + \frac{1}{S} \sum_{s \in G^s} g_s (\xi_s + \xi_s^*)
\]

subject to

\[
-\Psi^T w - \Phi \beta + \Phi \beta^* - \tau \xi - \xi^* - y \leq 0 \quad (\alpha) \]

\[
-\Psi^T w + \Phi \beta - \Phi \beta^* - \tau \xi - \xi^* - y \leq 0 \quad (\alpha^*)
\]

where symbols in brackets denote Lagrange multipliers, and \(\tau, g\) are vectors of length \(S\). The Wolfe-dual of this quadratic program reads:

\[
R_S = \max_{\alpha, \alpha^* \geq 0} y^T (\alpha - \alpha^*) - (\alpha^T - \alpha^*^T) K (\alpha - \alpha^*)
\]

subject to
\[ \Phi^T (\alpha - \alpha^*) \leq \frac{1}{\lambda} \tau \]  \hspace{1cm} (\beta)\\
\[ \Phi^T (\alpha - \alpha^*) \geq -\frac{1}{\lambda} \tau \]  \hspace{1cm} (\beta^*)\\
\[ i^T (\alpha + \alpha^*) \leq \frac{1}{\lambda} v \]  \hspace{1cm} (\epsilon)\\
\[ \alpha \leq \frac{1}{\lambda S} g \]  \hspace{1cm} (\xi)\\
\[ \alpha^* \leq \frac{1}{\lambda S} g \]  \hspace{1cm} (\xi^*)

where the matrix \( K = \Psi^T \Psi \) in the objective is the well-known kernel matrix. It may be noted that this matrix is square with the number of observations as dimension, and that, consequently, the number of eigenfunctions does not affect computations. This is critical in most applications of kernel learning, where in fact the matrix \( K \) is specified a priori as one with elements \( k(x_s, x_r) \) where \( k \) is a given kernel function i.e. a function such that \( K \) is symmetric and positive semidefinite. We note that from duality also follows that \( w = \Psi^T \alpha \) and the prediction at \( x_s \) is:

\[ \hat{y}(x_s) = \sum_{r \in G^0} k(x_r, x_s) (\alpha_r - \alpha_r^*) + \sum_j \phi_j(x_s) (\beta_j - \beta_j^*) , \]

which is a truncated form of the famous Representation Theorem by Kimeldorf and Wahba (1970), which establishes that also when the number of observations tends to infinity, the first term remains bounded, and that the form is sufficiently rich (flexible) to represent any true function \( y(x) \) exactly. Next, we refer to propositions 1 and 5 in Norkin and Keyzer (2005) that establish the key properties, presented here for the purely non-parametric form:

**Proposition 1 (Statistical properties of estimator).** Let Assumption 1 hold and consider the risk minimization problem \( R(f) \to \inf f \) over the class of functions that are continuous on the set \( X \). Then,

(a) for given \( x \), the minimum is achieved at the median \( \mu(x) \) of the conditional density \( \ell_x(y) \) and this solution is unique, and \( \mu(x) \) continuous.

(b) If \( \lambda \) is taken to depend on \( S \), the function \( \lambda(S) \) satisfies \( S \lambda^*(S) \geq S^\delta \), for any \( \delta > 0 \), then kernel minimizers \( f_S^{\lambda(S)} \) in (3.3a) converge to the true minimizer \( f^0 \) with probability one as \( S \to +\infty \).

We remark that in view of the continuity of \( \mu(x) \), we may limit attention to continuous kernel functions. Furthermore, the regularization, besides ensuring uniqueness of the optimum in (3.7) also maintains stability of the function as the sample expands, whereas without this regularization flexible function estimators necessarily suffers from ill-conditioning and overfitting and may, therefore, see their coefficients change dramatically from one sample to the next.

The median already is generally more robust than, say, the mean because it is insensitive to outliers. Yet, to obtain a robust estimator in the sense of Huber (1981), the loss function would have to be modified into the \( \epsilon \)-insensitive loss functions often used in kernel learning, such as
\[ c_\varepsilon (z, f(x)) = \max \{0, y-f(x) - \varepsilon\}. \] The corresponding $\varepsilon$-insensitive risk functional is \[ R_\varepsilon (f) = E_{x,y} c_\varepsilon (z, f(x)). \] It is obvious that $R_\varepsilon (f)$ uniformly converges to $R(f)$ as $\varepsilon \to 0$ but this is insufficient for establishing that minimizers $f_\varepsilon \in \arg \min_f R_\varepsilon (f)$ uniformly converge to $\mu = \arg \min_f R(f)$, because being an integral, the risk remains insensitive to change on an arbitrary small domain. Indeed, under this perturbation, only convergence in measure $P_j$ to this median as $\varepsilon \to 0$ can be established.

The semi-parametric form can be dealt with in the same way, recalling from section 3.1 that the median refers to the non-parametric term only i.e. to $y - \sum_j \phi_j(x) \beta$, and that the penalization $\tau_j$ on $\beta$ should also vanish with $S$.

**Window size**

So far, we took the kernel function as given, and indeed, the convergence properties hold for any fixed kernel function. However, as argued in Devroye et al. (1996, p. 423), in practical applications, it is essential to adapt the support of the kernel to the application at hand based on the (co-variance) of the observations and a coefficient, a window size $\theta^S$, that depends on the number of observations and the number of variables and such that the support of the density goes to 0 and consequently $k(x,x;\theta^S) \to 1$, as $S \to \infty$, ensuring that the density reduces to the empirical one. For example, if we use a normalized normal density as kernel function and if vector $x$ has $S_0$ elements, the kernel is the product of $T$ normalized Gaussian densities of the form

\[
\mu_j^S \exp \left( -\frac{1}{2} \left( \frac{x_{j_0} - x_{j_0}}{\theta^S \sigma_j^S} \right)^2 \right),
\]

where $S$ is the size of the sample used in the current estimation,\(^6\) $\sigma_j^S$ is an estimate of the standard deviation of variable $j$ and for window size (e.g. Haerdle, 1993):

\[
\theta^S = \left( \frac{4}{J+1} \right)^{1/(J+3)} S^{-1/(J+3)}, \quad \mu_j^S = 1/(\theta^S \sigma_j^S \sqrt{2\pi}).
\]

The window size essentially serves to determine the resolution of the calculation. A higher window size smoothen the profile of a response by attributing more weight to points that are distant in $x$.

**Regularization factor**

Norkin and Keyzer (2005) establish strong convergence for fixed window size and a regularization factor above a minimal threshold. Furthermore, the regularization factor that minimizes $E R\left( f_S^* \right)$ is shown to be:

\[
\lambda_S^* = \min \left\{ \frac{\kappa L}{S^{1/4}} \left\| f \right\|_H^2 \left( 18 \left(1 + \frac{C}{\kappa L}\right) + \frac{2}{\sqrt{S}} \right)^{1/2} \right\},
\]

where

\[^6\] Hence, under bagging it is the size of one bag.
(i) $S$ is the sample size used in the current round of estimation.

(ii) maximal kernel value: $|k(x,x)| \leq \kappa$ for all $x \in X$;

(iii) loss function $c(z,f)$ Lipschitz continuous uniformly in $z \in \mathbb{Z}$ for Lipschitz constant $L$, where $z$ is the error of fit;

(iv) maximal loss: $C = \sup_{z \in \mathbb{Z}} c(z,0); C < +\infty$;

(v) norm of true function: $\| f_0 \|_H^2 = \langle f_0, f_0 \rangle_H = \int \int f_0(x)k(x,x')f_0(x')dx$, where $f_0$ is the true function.

The constants $\kappa$, $L$, $C$ and $\| f_0 \|_H^2$ can be approximated as follows:

(ii) The kernel constant $\kappa$ can for Gaussian density be obtained as $\kappa = \frac{l}{(2\pi)^{J/2} \prod_j (\theta_j \sigma_j)}$, where $\theta_j$ is the window size and $\sigma_j$ the standard deviation of independent variable $j$. However, since $\kappa$ appears in ratios w.r.t. $C$ and $\| f_0 \|_H^2$, both linear in the kernel, and a close-to-zero standard deviation $\sigma_j$ or window size could create numerical difficulties, we choose to define a kernel without the denominator and hence equal to unity, noting that the denominator should be restored where the kernel is taken to be a density.

(iii) The Lipschitz constant $L$ in expression: $|\hat{c}(y_1 - f(x)) - \hat{c}(y_2 - f(x))| \leq L|y_1 - y_2|$, for the absolute value loss function, gives $\|y_1 - f(x)\|_\mathbb{R} - \|y_2 - f(x)\|_\mathbb{R} \leq L|y_1 - y_2|$. Assuming without loss of generality that $|y_1 - f(x)| \geq |y_2 - f(x)|$, the norm inequality $\|a + b\| \leq \|a\| + \|b\|$ implies that the condition holds for $L = 1$.

(iv) The maximal loss is the maximal value $\bar{y} = \max_x \sum \alpha_r k(x_r,x_x)$ in all samples.

(v) Though the constant $\| f \|_H^2 = \langle f, f \rangle_H = \alpha^T K \alpha$, is not observable, since this expression is nondecreasing in $\lambda$ we can bound it from above by choosing $\alpha$ obtained from prior estimation with $\lambda = \kappa$.

This suggests evaluating the regularization factor as:

$$\lambda_S = \min \left\{ \frac{\kappa}{S^{l/4} \alpha^T (\kappa K \alpha)^{-1} (\kappa K \alpha)^{-1}} \left( 18 \left( 1 + \frac{\bar{y}}{\kappa} \right) + \frac{2}{\sqrt{S}} \right)^{l/2} \right\},$$

where we remark that, unlike the window size that varies with the number of observations and independent variables only, the regularization factor is also specific to the dependent variable for which the regression is being conducted.

3.3 Accounting for district means in SV-regression

So far, we merely summarized the basic elements of kernel learning. We now suppose that we have in addition to the sample of grid cell data also a data set with district mean
\[ y^d = \frac{\sum_{s \in G} \delta_s^d f(x_s, \eta)}{\sum_{s \in G} \delta_s^d}, \quad (3.9b) \]

Hence, the census-weighted mean of the predictions should fit to the district data. We impose this as a (per unit) restriction on the kernel-learning program, i.e. by specifying an estimated value:

\[ y^d = \frac{\sum_{s \in G} \delta_s^d \left( \sum_i w_i \psi_i(x_s) + \sum_j \phi_j(x_s) \left( \beta_j - \beta_j^* \right) \right)}{\sum_{s \in G} \delta_s^d}, \quad (3.10) \]

for every district. We distinguish the case in which the eigenfunctions are known from the one in which the kernel function is known.

### 3.3.1 Choosing between treatment of district means as data or as constraints

Since (3.10) is linear in \((w, \beta - \beta^*)\), it would be possible to treat the district means as additional observations supplementing the survey data. However, when we seek to replace the eigenfunction by the kernel expression, the question arises as to whether the district means should enter the kernel functions. We only briefly review the issues this leads to since the approach to be implemented eventually will be a different one.

The estimated value of the district mean (3.9) can be written:

\[ y^d = \frac{\sum_{i} w_i \left[ \sum_{s \in G} \delta_s^d \psi_i(x_s) \right] + \sum_{j} \left( \beta_j - \beta_j^* \right) \left[ \sum_{s \in G} \phi_j(x_s) \right]}{\sum_{s \in G} \delta_s^d}, \quad (3.11) \]

we can incorporate it in program (3.7), (3.8) by redefining the matrices \(\Psi\), \(\Phi\) in dimensions \((S + D) \times I\), where \(D\) is the number of districts, and \((S + D) \times J\) for

\[ \Psi = \begin{bmatrix} \psi_i(x_r) \\ \sum_{s \in G} \delta_s^d \psi_i(x_s) \end{bmatrix}, \quad \text{and} \quad \Phi = \begin{bmatrix} \phi_j(x_r) \\ \sum_{s \in G} \delta_s^d \phi_j(x_s) \end{bmatrix}, \quad (3.12) \]

---

In applications, especially when the micro-data originate from a household survey, the function to be estimated will often express the dependent variable on a per capita, per household or per hectare basis and in estimation the error will be postulated in those terms as well. This can be accommodated by redefining the weight \(\delta_s^d\) in the district-constraint as measuring the number of people, households, and hectares, respectively, instead of being a 0-1 coefficient.
respectively, where the subscript \( r \) refers to the \( S \) data-points, and \( \delta_s^d = \frac{\delta_s^d}{\sum_{r \in G} \delta_r^d} \). This may also be interpreted as an aggregation from the (very large) \( \Psi \) and \( \Phi \) matrices defined as in (3.5) but now for all the grid cells in the region \( G \), using the aggregation matrix \( T \) with as non-zero elements, for the first \( S \) rows a unit element in the column that corresponds to the observation, and for the last \( D \) rows, unit elements in all columns that belong to the district of the row concerned. Then, the matrices of primal program (3.7a) can be evaluated as:

\[
\Psi = T\tilde{\Psi}, \quad \Phi = T\tilde{\Phi}. \tag{3.13}
\]

Clearly, this imposition of district means described here for kernel learning also applies to other estimation methods, for example to OLS as in (3.6).

Now, instead of primal (3.7a) we can also work with the dual (3.7b). However, for this, we must compute the kernel matrix \( \tilde{K} = \Psi \tilde{\Psi}^T \) for the full region and also the aggregation matrix \( T \) defined earlier, and on this basis evaluate the kernel matrix of program (3.7b) as:

\[
K = T\tilde{K}^T. \tag{3.14}
\]

Obviously, the matrix \( K \) is positive semidefinite as long as \( \tilde{K} \) has this property. For prediction, the functional form reads:

\[
\hat{y}_s = \sum_{r \in G} k(x_r, x_s) (\alpha_r - \alpha_r^*) + \sum_d \sum_{s', \in G} \hat{\delta}_{s'}^d k(x_{s'}, x_s) (\alpha_{d'}^* - \alpha_d^*) + \sum_j \phi_j(x_s) (\beta_j - \beta_j^*). \tag{3.15}
\]

Hence, we cannot use (3.8), since as every observation has a kernel term of its own the district means must be accounted for.

If instead of solving the primal problem with eigenfunctions, we start from an evaluation of the matrix \( \tilde{K} \) in (3.14) before solving (3.7b), the operation becomes computationally demanding both in estimation and prediction, in prediction because of the second term in (3.15) involves a large number of kernel evaluations, and in (3.14) because the kernel terms relating districts to one another requires the evaluation of all terms:

\[
K_{dd'} = \sum_s \hat{\delta}_{s'}^{d'} \sum_s \hat{\delta}_{s}^d k(x_{s'}, x_s), \tag{3.16}
\]

in the Gram-matrix, each of which would for a full coverage of the region by district amount to up to \( M^2 \) evaluations of the kernel function \( k \).

This is cumbersome, even when the evaluation of the Gram-matrix is done in dedicated (Fortran)-executables. We note in this connection that kernel functions that are concave for \( k \in [0, \varepsilon] \) have the property that the kernel value (similarity) between the means of two districts is greater than the sum of the pairwise kernels of these districts (disregarding the per capita division):

\[\text{The eigenfunction approach has definite advantages in that it may offer theoretical (micro-) foundations to the choice of the kernel function and offers a more stable parametrization, because the number of } \alpha \text{- coefficients required to represent the function can be controlled more directly (see Keyzer, 2004).}\]
which makes it possible to disregard pair of sites from districts that are sufficiently distant. Yet, for common forms this property will not be fulfilled in general. Consequently, pairwise tests cannot be circumvented. At the same time, we remark that the $K_{dd'}$ terms are only required for estimation.

To make applications for very large number of observations possible, we may recur to the calculation under OLS that operates on the normal form $\Phi^T y = \Phi^T \Phi \beta$, i.e. in dimension $S \times S$ and proceeds in two stages, with the terms of the normal form evaluated in Fortran for $\Phi$ defined as in (3.12), before the coefficients $\beta$ (unconstrained in sign) are obtained from the GAMS-program that minimizes the sum of absolute deviations from it, which is known to vanish in the optimum. The package in Appendix B has this option, also to allow for an easy comparison between OLS and SV-regression.

### 3.3.2 SV-regression, district means as constraints

Since treating the district observations as additional data is numerically cumbersome, we adopt the alternative approach of by substituting the kernel functions directly into the primal, and dealing directly with the now unconstrained coefficients $\alpha_r$ and $\beta_j$, and assigning the same $\xi_s$ to both constraints, since either one will be ineffective anyway. To account for possible imprecision in the district means we impose them as inequalities, with imprecision factor $\gamma$:

$$
\begin{align*}
\min_{\beta_j, \beta_j^*, \xi_s} & \frac{1}{2} \sum_{r \in G^o} \sum_{s \in G^o} \alpha_r k(x_r, x_s) + \sum_j \tau_j (\beta_j + \beta_j^*) + \nu \varepsilon + \frac{1}{S} \sum_{s \in G^o} \gamma \xi_s \\
\text{subject to} & \\
& y_s \leq \sum_{r \in G^o} k(x_s, x_r) \alpha_r + \sum_j \phi_j(x_s) (\beta_j + \beta_j^*) + \xi_s + \varepsilon \\
& y_s \geq \sum_{r \in G^o} k(x_s, x_r) \alpha_r + \sum_j \phi_j(x_s) (\beta_j - \beta_j^*) - \xi_s - \varepsilon, \quad s \in G^o. \\
& -\gamma y^d \leq \sum_{r \in G^o} \left[ \sum_{s \in G^o} \hat{\delta}_s k(x_r, x_s) \right] \alpha_r + \sum_j \left[ \sum_{s \in G^o} \hat{\delta}_s \phi_j(x_s) \right] (\beta_j - \beta_j^*) \leq \gamma y^d,
\end{align*}
$$

where the terms in square brackets of the district constraints define matrices indicating that before solving the program, all census information by summation can be compressed into constraints with $2S + J$ unknown and a constraint matrix of dimension $(2S + D) \times (2S + D + J)$.

Moreover, the computational burden in computing the terms in square brackets seems manageable, and easier than for the alternative formulation in (3.4) as product terms of dimension $D \times D$ whose preparation requires calculations on $M \times M$ can be avoided.

Furthermore, the interpretation of district data as fixed, i.e. sample independent, constraints of the risk minimization problem, has the added advantage that the statistical setting becomes more transparent, as the data now merely consists of the survey sample, ensuring that the estimators inherit all properties of SV-regression, for instance about strong convergence with rising sample size.

The package in Appendix B implements (3.18) as well as the variant with given eigenfunctions, i.e. with non-parametric term $\sum_i w_i \psi_i(x_s)$ replacing $\sum_{r \in G^o} k(x_s, x_r) \alpha_r$. 

3.3.3 Non-linearity in $\beta$; treatment of latent variables

The SV-literature generally limits attention to functional forms that are linear in $(\alpha, \beta)$. Indeed, we have seen that for $\alpha$ linearity is not a limitation. Assuming linearity w.r.t. $\beta$ is practical as it makes it possible to rely on quadratic programming, which seems also essential in obtaining the Wolfe-dual (3.7b). However, as indicated in program (3.18), the kernel form can be introduced directly into the primal problem. Moreover, Proposition 1 shows that the linearity of the constraints and in fact the specific functional form of $\xi_s(\alpha, \beta)$ does not matter as long as the program is convex in $(\alpha, \beta)$. Hence, the quadratic programming form is not essential and could be dispensed of.

This generalization is not needed to represent any specific response to $x$, since the expansion of terms $\phi_j(x)$ can be of arbitrary length and restrictions on $\beta$ can be imposed to fit any function, as is readily understood from the similarity between eigenfunction terms $w_i \psi_i(x)$ and the parametric terms. Yet, the nonlinearity in $\beta$ will naturally enter when the parametric term is to be derived from some process-based model or from a micro-economic optimization.

Regarding convexity, the presence of upper as well as lower bounds in (3.18) is problematic for any nonlinear form, as concavity with respect to one bound amounts to convexity with respect to the other, while both are supposed to be binding each for a different subset of observations. Nonetheless, various problems can be accommodated.

As an illustration, we look into the standard consumer problem of utility maximization subject to a budget constraint $\max_{c_h \geq 0} \{ u(c_1, ..., c_H; \beta) \} | \sum p_h c_h = m \}$ where $u$ is a given, differentiable and concave parametric form of the (direct) utility function, and where dependent variable $c_h$ is the consumed quantity of commodity $h$, while $p_h$ is the given price of commodity and $m$ the given total expenditure. The regression problem is to estimate parameter $\beta$ of the demand functions $c_h \left( \frac{1}{m_s} p_{l_s}, ..., \frac{1}{m_s} p_{H_s}; \beta \right)$ associated to the utility maximization problem on the basis of observations $y_{hs} = c_{hs}$ and $x_{hs} = \frac{1}{m_s} p_{hs}$, $s = l, ..., S$. For this, we rely on the first-order conditions:

$$
\begin{align*}
\frac{u_h'(c_{l_s}, ..., c_{H_s}; \beta)}{\beta} &= \pi_s p_{hs}, \\
\sum_h p_{hs} c_{hs} &= m_s, \\
c_{hs} &\geq 0
\end{align*}
$$

where $\pi_s$ is the (unobserved or latent) Lagrange multiplier of the budget constraint and $u_h'$ denotes the partial derivative of utility w.r.t. $c_h$. Note that except for this derivative, the conditions are linear in unknowns $c_{hs}$ and $\pi_s$ and can, therefore, readily be inserted in the quadratic program, which illustrates that the quadratic programming format may be particularly suited for dealing with latent variables such as Lagrange multipliers. Furthermore, for some well known closed forms of direct utility the derivative is linear in $\beta$ as well: e.g. for Cobb Douglas
\[ u = \beta_h \ln c_h, \text{ or addilog with given exponents } u = \sum_h \beta_h (c_h)^{y_h} \] but more general forms tend to involve multiplicative terms \( \beta_h \beta_h' \), that result in non-convex constraints in \( \beta \). Yet, when sufficient data are available the flexible form \( u = \sum_{h,j} \beta_{hj} \phi_{hj} (c) \) can for concave functions \( \phi_{hj} \) and constraints \( \beta_{hj} \geq 0 \) accommodate any concave utility, and fits well within the quadratic programming format, albeit that the demand functions for all commodities should be estimated in a single program.

We conclude that the introduction of nonlinearity in \( \beta \) has to be dealt with on a case-by-case basis. The GAMS-program SVvBAG.GMS can readily be tailored to this purpose but the convergence of the resulting possibly non-convex program has to be monitored closely.

### 3.4 Mollifier (kernel smoothing), no processing of district means

Mollification (kernel smoothing, see e.g. Haerdle, 1993) also uses the kernel function as a proximity measure but whereas so far, the kernel functions were merely required to generate a positive semidefinite Gram matrix \( K \), the mollifier or kernel smoothing approach assumes in addition that the kernel function \( k(x_s, x; \theta^S) \) is non-negative and measurable on \( X \) with unit integral and with compact support. Hence, for given sample of size \( S \), the kernel value \( k(x_s, x; \theta^S) \) denotes the likelihood of \( y \) observed at \( x \) being equal to \( y_s \), possibly estimated itself by kernel methods in an earlier round but definitely characterized by a window size as discussed in section 3.2.

Kernel smoothing is an interpolation or function averaging technique that lays a “smooth blanket” over the data points (spikes) of the empirical distribution. In the regular situation with survey observations only, the mollified function (Nadaraya-Watson regression) reads:

\[
\hat{y}_s(\mathbf{x}_s) = \sum_r P_r(x_r, x_s) y_r,
\]

for probability calculated as

\[
P_r(x_r, x_s) = \frac{k(x_r, x_s; \theta^S)}{\sum_{s' \in G^s} k(x_{s'}, x_s; \theta^S)}.
\]

In section 3.6 we return to the use of kernel smoothing for different purposes.

### 3.5 Stability of prediction: mapping of errors and bagging

The non-parametric term of SV-regression can be interpreted as an estimate of the error in prediction of the parametric part relative to the best possible function. This error can be evaluated at every point of the map or census. By contrast the remaining error \( (\xi_s - \xi_s^e) \) is after the regression only available at data points of the survey. It enters the empirical risk criterion and gives an estimate as to how well any function could at the given regularization \( \lambda \) and soft-margin
represent the given data, since in principle an infinity of $y$-values, as represented by the conditional density $g(y \mid x)$ could correspond to every given $x$.

In addition, the stability of these predictions can be measured with respect to the available sample. Here simulation approaches suggest themselves to supplement the information on errors from the full sample. As simulation approach, we consider bagging since it can include most of the competing approaches (leave-one-out, jackknife, bootstrap) as special cases. In addition, the stability of these predictions can be measured.

Bagging is a technique of stochastic simulation that considers a series of subsets of the data set obtained through subdivision or sampling (with or without replacement) and evaluates the variability of predictions as well as of $\beta$-coefficients across samples, for example by evaluating the standard deviation of predictions. For $\alpha$-coefficients this is less meaningful as these are heavily dependent on the sample. Bagging is also practical when the sample size makes it difficult to solve the quadratic program. See Appendix B for further details on the implementation.

### 3.6 Probability and likelihood bounds

Bagging and other simulation techniques only report on the stability with respect to the available sample, not on the spread of the underlying conditional density around the estimated function. If a good representation of this conditional density were available, estimation of the function would become a purely arithmetic exercise. However, no such representation is available and the statistical learning approach precisely avoids postulating any closed form of $g(y \mid x)$. Moreover, sample size will generally be too small to invoke asymptotic properties. Finally, the information in the census and grid map through the postulated regression function implicitly define an empirical distribution of $(y,x)$ as mapped out in the prediction step that would conflict with any a priori chosen closed form.

Hence, a tentative approach has to be settled for. Here we opt for confronting the prediction over census and grid map with a smoothed form of the empirical conditional distribution, to be obtained by kernel smoothing. Though the quality of this distribution necessarily is questionable, this procedure enables us to keep the representation of the density fully separate from the function estimation, and hence to avoid using up additional degrees of freedom.

As in kernel smoothing, we now assume that the kernel function $k(x_s,x)$ is non-negative and measurable on $X$ with unit integral and compact support. The kernel function is also characterized by a window size $\theta^S$ such that the support of the density goes to $\theta$ and, consequently, $k(x,x,\theta^S) \to 1$, as $S \to \infty$, ensuring that the density eventually reduces to the empirical one. Yet, we drop the window size for notational convenience.

Hence, for given sample of size $S$, $k(x_s,x)$ is taken to denote the likelihood of $y$ observed at $x$ being equal to $y_s$. This is in line with the mollifier formulation, which directly calculates the conditional mean of the associated conditional probability, and not in contradiction with SV-regression, which abstracts from specifying directly the conditional density $g(y \mid x)$. Indeed, under the stated assumptions on the window size vanishing as $S \to \infty$, the construction is merely a device to characterize, for given $S$, the density at points $x$, where no data are available. The
device may be plausible in that it follows the logic of kernel learning attributing higher likelihood density to nearby observations $y_s$. It does not require constructing new observations $y_s$ from any previously estimated density. Yet, the kernel used could be estimated itself by kernel methods at an earlier stage.\(^9\)

Consequently, if we further assume that $\sum_{r=1}^S k(x_r, x) > 0$, we can evaluate the conditional probability:

$$P(y \leq y \leq \bar{y} | x) = \frac{\frac{1}{S} \sum_{r | y \leq y \leq \bar{y}} k(x_r, x)}{\frac{1}{S} \sum_r k(x_r, x)}.$$

(3.21)

which is a measure of fit at $x$.

In fact, the denominator points to the likelihood of having any observation at $x$. Hence, for given regression function, a possible measure of accuracy of the prediction could be the conditional probability of $y$ at $x$ falling with a margin $\beta$ of the estimated value $\hat{y}(x)$:

$$L_\beta^S (x) = P(|y - \hat{y}(x)| \leq \beta \hat{y}(x) | x) = \frac{\frac{1}{S} \sum_{r | y \leq \hat{y}(x) \leq \bar{y} \leq (1+\beta) \hat{y}(x)} k(x_r, x)}{\frac{1}{S} \sum_r k(x_r, x)}$$

(3.22)

or alternatively, evaluate the conditional likelihood of $y$ at $x$ falling within the given range:

$$L_\beta^S (x) = \frac{1}{S} \sum_{r | y \leq \hat{y}(x) \leq \bar{y} \leq (1+\beta) \hat{y}(x)} k(x_r, x).$$

(3.23)

The latter has the advantage that it has a high value at points close to sample observations, where the spread of the conditional density is low. A high value signals that the prediction is based on a good fraction of the data sample and that there are relevant data points with $y$-values in the $\beta$-range.

The calculation can be conducted in parallel for deviations from the parametric part of the model, since the function is taken as given independent of the way in which the function was estimated. Be this as it may, for samples of small to moderate size these calculations depend heavily on the kernel function that is assumed to represent the conditional density, and only as the sample goes to infinity and the window size drops to zero, it converges to the true density. The basic difficulty is that while it is possible to construct a histogram of $y$ as an empirical probability distribution without requiring any model, for a conditional probability distribution a model is needed, for

---

\(^9\) Alternatively, the conditional density itself could be estimated at that stage, with the advantage that it could incorporate various properties, such as smoothness, which the empirical density does not possess and could be expressed in more compact form, say, as a histogram with a limited number of classes. However, if this is possible, the question becomes why not to use this density for estimating the regression function itself, or even why not to derive the conditional median purely arithmetically from this density. One possible answer could be that the main estimation is used to obtain a “good” parametric part, and that the pre-estimation prepares for this by adding qualitative information to the necessarily sparse empirical density. At any rate, this density would generate a continuum of observation, and, therefore, require an adaptive stochastic optimization algorithm. These are issues for further research.
instance the kernel forms (3.21)-(3.23). If we are willing to accept these probability expressions as data, it becomes possible to conduct regressions on them, say, by conducting an SV-regression on an inverse logistic function. Yet, the approach is questionable because the probability calculation may too strongly depend on the kernel function assumed. In sum, the approach is tentative and stands to be replaced by a better one.

3.7 Census weights on survey observations

The census information can, for small to moderately sized samples, be used to attribute weights to data in the sample. For an iid sample, the empirical risk from a survey sample

\[ R^e = \frac{1}{S} \sum_{s=1}^{S} g_s \xi_s, \] (3.24)

should be computed with unity weights \( g_s = 1 \). However, even when data are sampled from a full list of households, individuals or grid cells, the measure of the characteristics sampled may not be iid. Consequently, the empirical distribution of \( x \) will with rising sample size only very slowly converge to the distribution in the population. Now the information at population level (census or map) can be used to redress the sample through weights, attributing in the SV-regression program (3.7), a high weight to observations near to clusters in the census:

\[ g_s = \frac{1}{M} \sum_{r \in G} k(x_r, x_s; \Theta^S), \quad s = 1, \ldots, S, \] (3.25)

where \( k(\cdot, \cdot, \cdot) \) is taken to be a density. Hence, if the census had uniform measure over \( X \), for census size \( M \) large enough, the weight would usually come close to unity (irrespective of the distribution of the survey). In the more common case that it has a non-uniform measure, the weight will be relatively higher for data cluster. Furthermore, for \( S \to \infty \), the window size will become zero and the weight move to unity, and in fact for large \( S \), the weight can be put to unity anyway as the sample will come close to the census (since the census was taken to be a true representation of the population).

3.8 Endogeneity and Autocorrelation

We recall that Assumption 1, postulates a conditional density \( g(y | x) \) and as far as the error in regression is concerned only requires under (d) that on the set \( X \), the expected error or disturbance function \( \Delta(x) = E_{y|x} [y - f(x)] \) is supposed to be continuous in \( x \) wherever \( f(x) \) is continuous. The strong consistency of SV-regression stated in Proposition 1 does not depend on any other assumption about the error.

In fact, apart from the fact that the density might be varying in time, there is no possibility for misspecification in this distribution free setting. Any missing factors among the independent variables will be reflected in the shape of the conditional density. Specifically, there is no need to assume zero covariance between the independent variables and the error term. On the contrary, as was mentioned before the non-parametric term can be interpreted as an error function that depends on \( x \). Hence, endogeneity is not an issue in this respect.
Yet, the more fundamental issue remains that conditioning is not the same as control. Conditioning says nothing about the ease of modifying $x$. For example, if observations are cross-section this $x$ may refer to immutable variables. Econometrics in such a case suggests instrumentalization on variables $z$ (e.g. Greene, 2000). Rather than using the full domain $X$ in the range of the conditional density $g(x|z)$, instrumentalization restricts the domain $X$ to the range of the regression function $x(z)$ but this only relates to the use of the function in prediction. As far as the convergence of the regression function $y(x)$ is concerned, there is under SV-regression no clear gain in limiting attention to this subset of $X$ but it may be possible through it to limit the spread of the conditional density around the conditional median.

Similarly, extending the list of explanatory variables, say, to include $z$, may be an option, provided a suitable kernel function is available.

Autocorrelation

Also, the conditions at neighboring sites (and recent time periods) may be seen as additional explanatory factors, which could justify accounting for autocorrelation both in errors and in dependent variables. We now discuss to which extent autocorrelation concepts of econometrics (e.g. Davidson and MacKinnon, 2004) can be accounted for in the SV-framework.

First, the kernel function uses information on neighbors and through the non-parametric term actually allows for observations from neighbors to affect the prediction. Note that since no neighbors can be excluded, the kernel is less suited to represent hierarchical relationships such as physical flows or dynamics. Yet, the eventual estimate of the conditional median will only deal with the on-site independent variables.

Second, if an effect from neighbors is to be included in the median, the vector $x$ can be extended and taken to include conditions at neighboring sites. These conditions can then enter the kernel functions that may exhibit an appropriate weighting for distance to the site under consideration as well as the necessary hierarchy. Indeed, this can be used to define special eigenfunctions (e.g. wavelets) as building blocks of spatial or dynamic patterns representing the solution of the difference equation in a flexible way, rather than the difference equation itself.

Third, whereas the first and second channel refer to an effect via an observed independent variable, one might also envisage an effect via an unobserved one, which in the SV-framework finds expression in the spread of the conditional density of $y$ and hence in the error term that only depends on the $x$ at the site. A parametric representation of autocorrelation in these effects amounts to taking the error at site $s$ to depend on errors at neighboring sites as well as on a local error. For example, suppose that for (e.g. spatially or temporally) uncorrelated errors $(\zeta - \zeta^*)$, the actual errors obey the linear relationship:

$$\xi - \xi^* = A(\zeta - \zeta^*) + (\zeta - \zeta^*).$$

(3.26)

where the matrix $A$ is taken to have been estimated by some dedicated technique (e.g. kriging) and $(I - A)$ is non-singular. The zeroes in the matrix $A$ now make it possible to depict hierarchies. The difference $(\zeta - \zeta^*)$ in fact becomes an unobserved dependent variable, and $(\zeta - \zeta^*)$ an unobserved dependent variable, and the functional specification extends to:
\[ \hat{y}(x_s, (\xi_1 - \xi^*)_s, ..., (\xi_S - \xi^*_S)) = f(x_s) + \sum_{r \neq s} A_{sr} (\xi_r - \xi^*_r). \]  

(3.27)

Yet, our aim remains the estimation of \( f(x) \). This means that we should take the second term on the right-hand side in (3.25) to the left, i.e. subtract it from the observation \( y_s \), and treat the difference as data for SV-regression. However, since \((\xi - \xi^*)\) is unobserved, we cannot do this prior to regression and must include (3.26) within the estimation. Now for \( C = (I - A)^{-1} \), we have \( \xi = C \xi^* \), \( \xi^* = C \xi^* \), and, therefore, the modified SV-regression problem (where we abstract from the district constrain, for notational ease):

\[
R_S = \min_{\alpha; \beta, \beta^*; \xi, \xi^*; \alpha; \beta, \beta^*; \xi, \xi^*} \frac{\lambda}{2} \alpha^T K \alpha + \tau^T \beta + \tau^T \beta^* + \nu \epsilon + \frac{1}{S} g^T \xi + \frac{1}{S} g^T \xi^* \\
\text{subject to} \\
-K \alpha - \Phi \beta + \Phi \beta^* - \epsilon \xi - C \xi^* + y \leq 0 \\
K \alpha + \Phi \beta - \Phi \beta^* - \epsilon \xi^* - y \leq 0
\]  

(3.28)

where the loss function is expressed in terms of the underlying not autocorrelated error \( \xi - \xi^* \). In fact, we may in (3.26) redefine the weight as \( \tilde{g} = M^{-1} g = (I - A)g \), the program reduces to the usual form:

\[
R_S = \min_{\beta, \beta^*; \xi, \xi^*; \alpha; \beta, \beta^*; \xi, \xi^*} \frac{\lambda}{2} \alpha^T K \alpha + \tau^T \beta + \tau^T \beta^* + \nu \epsilon + \frac{1}{S} g^T \xi \\
\text{subject to} \\
-K \alpha - \Phi \beta + \Phi \beta^* - \epsilon \xi - \xi + y \leq 0 \\
K \alpha + \Phi \beta - \Phi \beta^* - \epsilon \xi^* - y \leq 0
\]  

(3.29)

dispensing of the non-singularity requirement on \((I - A)\) and illustrating that (non-unitary) weighted SV-regression can account for autocorrelation patterns. We also remark that the soft margin inactivates observations that lie close to the regression curve, irrespective of their number. This reduces the effect of e.g. spatially or temporally clustered data.

Fourth, the intensity of error autocorrelation could be varied by iterative calls of the SV-program while adjusting either the window size \( \theta^S \) in the kernel function or a weight \( \omega \) in \((I - \omega A)\) so as to minimize the empirical risk. We also remark that the empirical risk \( R_S \) in program (3.29) is convex in \( A \). Consequently, it would be possible to nest this program within a further risk minimization with adjustment of these coefficients. This is a topic for further research.

Finally, in addition to allowing for error autocorrelation, it would be possible to treat the dependent variable of neighboring periods or neighboring sites as explanatory variables. The well-known difficulty is then that biases are introduced if these dependent variables are simply treated like independent ones. The approach of modeling the interdependences directly through the constraints can be effective here as well. For example, consider the model with a \( \tau \)-period distributed lag, linear in dependent variables:

\[
y_t = \sum_{\tau=1}^\tau y_{t-\tau} \gamma_{\tau} + \sum_i w_i \psi_i(x_i) + \sum_j \phi_j(x_j) (\beta_j - \beta^*_j) + \eta_t
\]  

(3.30)
Treating $y_{t-\tau}$ as given is unwarranted since this variable is endogenous. Yet, treating it as endogenous, introduces a non-convexity via the products $y_{i-\tau}\gamma_{i}$ of two choice variables. Such non-convexity is to be avoided not only because it makes computation of the risk minimizing optimum more complex but also because it undermines the convergence of function estimates. Three approaches are available. The first is differencing, i.e. postulating a model other than (3.30) written in $\Delta$- or in $\Delta\log$-form in which the dependency of the right-hand side is considered to have been eliminated, provided some stationarity test were passed. The second approach is concerned with the error-free version of (3.30):

$$\hat{y}_t = \sum_{\tau=1}^{\tau} \hat{y}_{t-\tau} \gamma_{\tau} + \sum_i w_i \psi_i(x_i) + \sum_j \phi_j(x_j)(\beta_j - \beta^*_j)$$

(3.31)

for given $\hat{y}_{t-\tau}$ at $t = 0$, to be confronted with the observations $y_t$ in every period, up to the soft-margin deviation. The $\gamma_{\tau}$-coefficients are obtained from some estimation procedure conducted prior to estimation, in which case only the dependency of $y_{t-\tau}$ remains. Such two-stage procedures are common in econometrics but seem at odds with the SV-approach that, to maintain strong convergence without unnecessary qualifications, has to insist on solving the risk minimization problem in its pure form. The third approach also operates with (3.31) but would proceed by successive rounds of gradient-based adjustments of $\gamma$ treated as given in the standard SV-risk minimization (and similarly for the coefficients of the autocorrelation of the error). However, in the SV-setting, this will create a non-linearity w.r.t. $\gamma$ through exponentiation and hence a non-convexity, since the SV-form necessarily involves upper as well as lower bounds. Yet, scanning on a scalar $\omega$ (or $\gamma_j$ in case of single period lag $\tau = 1$) to adjust the speed of decline of $\gamma_{\tau}$ with $\tau$ is always possible. Alternatively, we can solve in a first round an SV-regression with (3.30), followed by a second round in which $\gamma$ is re-estimated with (3.31), i.e. with a product form, but is kept within an interval around the original value. Keeping the interval sufficiently narrow will avoid the non-convexity to occur within the feasible set. To sum up, it is possible to account for autocorrelation within the SV-setting but non-convexities will arise as one attempts to estimate $\gamma$-coefficients simultaneously with $\alpha$ and $\beta$. 
4. Classification

4.1 Short summary on classification

4.1.1 Rule-based classifiers

So far, the dependent variables were taken to be real-valued. No particular attention was given to their range, as the kernel techniques considered are by their very nature distribution-free and hence unlike, say, maximum likelihood estimation under normality assumptions, not requiring the support of the conditional density \( g(y|x) \) to be unrestricted. Here we turn to the case of discrete-valued dependent variables (cases (3)-(6) in the introduction) that adopt values \( h = 1, \ldots, H \), while \( h = 0 \) refers to non-classification. Independent variables can be real- as well as discrete-valued.

(a) Expert-functions. These are classification functions fully specified by experts. Such functions can be can be stated here more naturally than for real-valued dependent variables, because we only aim at specifying a binary choice. GIS-applications often rely on them when physical objects have to be recognized. They often have a tree-structure, whereby the classification decides say, that an animal without a trump cannot be an elephant, and quits further investigation. Yet, even expert- functions tend to contain some parameters such as window size that may have to be adjusted prior to prediction.

(b) Clustering. This widely used technique compares an independent variable observation \( x \), with given reference values \( a_j \) and chooses the \( j \)-value for which

\[
\min_j D(x,a_j) = \arg \min_j D(x,a_j),
\]

where \( D \) is some distance function.

(c) Stone’s nearest neighbor: unlike expert functions, nearest neighbor classification requires iid observation pairs \((x_s,y_s)\) for binary valued \( y_s \), e.g. \( y_s = -1,1 \) but it does not conduct any parameter estimation prior to prediction. The technique chooses according to the highest the average vote among the \( N^S \) nearest neighbors of \( x \), i.e. the \( N^S \) observations \( x_s \) closest to \( x \). This defines an \( S \)-dependent decision rule \( c^S(x) \) for which it can be proved that for \( S \to \infty \) it is strongly consistent, i.e. converges with probability one to the rule with minimal probability of error, provided \( N^S \to \infty \) and \( N^S / S \to 0 \), moreover, for \( N^S = 1 \), the largest expected value of the probability of error is not larger than twice the maximal value (results by Stone, 1977 and Cover-Hart, 1967, see Devroye et al., 1996). See also section 4.4 below.

The nearest neighbor algorithm illustrates that unlike for real-valued dependent variables, binary choice supports strong convergence with simple algorithms that require survey data but do not need parameter estimation.

4.1.2 SV-classification

Estimation algorithms based on empirical risk minimization tend to perform better for smaller samples. Therefore, we consider as an alternative to nearest neighbor classification, various techniques of SV-classification. We further discuss kernel smoothing that is also a form of kernel learning, since it makes use of the survey data and a kernel representation even though it is not
risk minimizing. Because of the discrete nature of the dependent variables, OLS is not an option here but as mentioned earlier SV-classification could readily be implemented with least squares (Györfi et al., 2003) instead of the absolute value minimization to be considered here.

Referring to the notation of Section 3, we now take the independent variables to define a (semiparametric) discrete choice function. We allow for two options, one with and one without a “zero” or unclassified outcome:

\[
\begin{align*}
    c(x) &= \arg\max_{h \in \{1, \ldots, H\}} f_h(x), \quad \text{(4.1a)} \\
    c(x) &= \arg\max_{h \in \{1, \ldots, H\}} f_h(x) \quad \text{if } \max_{h \in \{1, \ldots, H\}} f_h(x) > 0 \text{ and } 0 \text{ otherwise}, \quad \text{(4.1b)}
\end{align*}
\]

respectively, for \( f_h(x) = \sum_i \psi_i(x_s) w_{ih} + \sum_j \phi_{jh}(x_s) \beta_{jh} \), where the parametric term includes a constant: \( \phi_{jh}(x_s) = 1 \) for \( j = 1 \) and every \( h \).

Observations of dependent variables are on the class \( h_s \in \{0, 1, \ldots, H\} \) corresponding to \( x_s \). Hence, for every \( s \), we can specify the binary values

\[
y_{hs} = 1 \quad \text{if } h = h_s \text{ and } -1 \text{ otherwise}, \quad \text{(4.2)}
\]

with \( 1 \) referring to \( s \) belonging to class \( h \), and \(-1\) to \( s \) not belonging to this class. Although the requirement can be circumvented, it is natural to assume that at every \( s \), \( y_{hs} = 1 \) holds for one and only one \( h \). All classifiers considered here have in common that they can be cast in this setting. They include:

(i) **Single class problem**: all \( y \)-values are equal to one, and the classification is only used to select observations, through its soft-margin.

(ii) **Binary class problem**: finding \( f \) for a single \( h \).

(iii) **Multiple class problem**: solving (4.1) for multiple \( h \). Here we distinguish class wise estimation from joint estimation that derives all class wise functions from a common optimizing model. We also discuss briefly how to deal with multistage decisions.

(iv) **Ranking**: as multiple class but with \( f_h(x) = \min(f(x) - \bar{f}_h, \bar{f}_{h+1} - f(x)) \), i.e. derived from simultaneous estimation of \( f \) and the bounds \( \bar{f}_h \) such that \( \bar{f}_h \leq \bar{f}_{h+1} \).

Regarding the estimation technique, we allow for:

---

10 Maintaining concavity or monotonicity of the parametric term may require imposing non-negativity constraints on the coefficients \( \beta \). Furthermore, expert functions can be incorporated by keeping a \( \beta \)-coefficient fixed at unit level. Furthermore, technologically specified expert functions can be incorporated via a single term without any \( \beta \)-coefficient. This should be controlled directly in the GAMS-estimation program CLvBag.gms.

11 We remark that this assignment differs from the common practice of logit regression in econometrics where \( 0 \) usually is the value assigned to non-occurrence, and the estimation only considers occurrence. Kernel learning treats occurrence and non-occurrence symmetrically.
(a) SV-classification, with either eigenfunctions or kernel functions as given, while incorporating district information.
(b) Nearest neighbor classification: only for binary and multiple class problems, and without inclusion of district data.
(c) Kernel smoothing (mollifier) classification: only for binary and multiple class problems, and without inclusion of district data.

With the exception of the nearest neighbor, the approaches considered as in SV-regression all operate on a conditional density \( g(\tilde{y} \mid x) \) in real-valued variables, except that \( \tilde{y} \) is now latent as opposed to being observed directly, and has to be inferred from observations \( y \). Consequently, SV-classification can also minimize a risk criterion expressed in real-valued variables and the difference only resides in the nature of the constraints on this minimization that are imposed by the observations.

Hence, SV-classification is a form of kernel learning and we can refer back to section 3.2.2 for a brief discussion of the underlying principles, and to sections 3.6-3.8 for the discussion on the mapping of errors and stability of predictions (bagging), the calculation of census based weights, and the treatment of autocorrelation, respectively.

### 4.2 SV-classifiers of binary, single and multiple class choice

#### 4.2.1 Binary choice

SV-classifiers estimate a different binary choice function \( f_h \) separately for every class. The key difference from the SV-regression of Section 3 is that while whereas in SV-regression the function should come close to the observed value of the dependent variable, in SV-classification the binary observations (4.2) on \( y \) actually express that the function \( f_h \) should be non-negative and non-positive, respectively. This can represented by specifying constraints in the form \( y_{hs} f_h \leq \xi_s \), where \( \xi_s \) is the regression error, i.e. by multiplying the function with \( y_{hs} \) on the left-hand side of a constraint with a \( \leq \) sign.\(^\text{12}\)

Hence, if one also introduces an endogenous threshold \( \sigma \) to promote strict separation of both cases, the primal program can, for the case of given eigenfunctions, be written, dropping the subscript \( h \) for notational convenience:

\[
\begin{align*}
\min_{\beta_j, \beta_j^*, \xi_s \geq 0, \sigma \geq 0, w_i} & \quad \lambda \frac{1}{2} \sum_i w_i^2 + \sum_j \tau_j (\beta_j + \beta_j^*) - v \sigma + \frac{1}{S} \sum_s g_s \xi_s \\
\text{subject to} & \quad y_s \left( \sum_i w_i \psi_i (x_s) + \sum_j \phi_j (x_s) (\beta_j - \beta_j^*) \right) \geq \sigma - \xi_s
\end{align*}
\]

\(^{12}\) It would in principle be possible to conduct SV-regression on the discrete variables just like for real variables. This has the advantage of leading to probability estimates as predicted dependent variables. Yet, disadvantages are: (i) the kernel expansion may because of negative \( \alpha \)-coefficients lead to negative predictions; (ii) estimates in the intermediate range of \( y \) between 0 and 1 are likely to be unreliable as there are no observation while those are precisely the ones that trigger the decision in prediction.
with \( \tau_j = 0 \), to which corresponds the classification decision \( c(x) = \text{sgn}(f(x)) \), \( (\tau_j = 0 \) is admissible for all \( j \)). A key difference from SV-regression is that the soft margin \( \sigma \) is now subtracted in the objective where it is added in SV-regression. Hence, \( v \) is not a penalization but a reward and for \( v \) large enough the program will become unbounded. Clearly, we take \( v \) to be sufficiently small to avoid this and in practice it is practical to place some upper bound on \( \sigma \) that should be ineffective in the optimum.

As for SV-regression, there is a dual form with kernels derived from the given eigenfunctions or postulated directly.

\[
R_S = \min_{\beta, \beta^*, \sigma, \xi, \omega} \frac{1}{2} \omega^T \omega + \frac{1}{\lambda} \tau^T \beta + \frac{1}{\alpha} \tau^T \beta^* - \frac{1}{\lambda} v \sigma + \frac{1}{\lambda S} g^T \xi
\]

subject to
\[
Y(\Psi w + \Phi \beta - \Phi \beta^*) \geq \sigma \tau - \xi \quad (\alpha)
\]

with dual

\[
R_S = \min_{\alpha, \xi, \omega} \alpha^T Y K \alpha
\]

subject to
\[
\Phi^T Y \alpha \leq \tau \quad (\beta)
\]
\[
\Phi^T Y \alpha \geq -\tau \quad (\beta^*)
\]
\[
\tau^T \alpha \geq v \quad (\sigma)
\]
\[
\alpha \leq \frac{1}{\lambda S} g \quad (\xi)
\]

where the regression constant \( \phi_j(x) = 1 \) for \( j = 1 \) ensures that \( y^T \alpha = 0 \) holds.\(^{13}\) Alternatively, we can as before substitute the dual form into (4.4a) and work with:

\[
R_S = \min_{\alpha, \beta, \beta^*, \sigma, \xi, \omega} \frac{1}{2} \alpha^T Y K \alpha + \tau^T \beta + \tau^T \beta^* - v \sigma + \frac{1}{S} g^T \xi
\]

subject to
\[
Y(K \alpha + \Phi \beta - \Phi \beta^*) \geq \sigma \tau - \xi .
\]

We note that since it is possible to restrict \( (\beta - \beta^*) \) through additional bounds and constraints, and \( \alpha \) is non-negative with given bounds apparent in (4.4b), the range of the estimated function can, for nonnegative kernel functions, be confined as well. For a discussion on the scope for

\(^{13}\) As for SV-regression, actual calculations will be conducted after division of all coefficients of the primal objective by the regularization factor \( \lambda \). Recall also that \( \tau_j \) should vanish with rising \( S \) and could be dropped from the outset.
incorporating non-linearity in \( \beta \) and the treatment of latent variables, we refer to section 3.3.3 above.

It is important to remark as well that SV-classification (4.5) only differs from SV-regression in the constraints. The risk criterion is of the same form enabling classification-methods to inherit various convergence properties albeit that the identifiability of individual coefficients is different. Moreover, this indicates that in case of simultaneous estimation of various functions, classification restrictions can be combined with regression constraints to constitute hybrids of various kinds.\(^\text{14}\)

**Relation to logit and probit estimation**

Logit and probit estimation (see e.g. Greene, 2000) are parametric techniques that choose a value for parameter \( \beta \) so as to maximize the log-likelihood of the sample,

\[
\ln L = \frac{1}{S} \sum_{s} \left[ q_{s} \ln P_{s} + (1 - q_{s}) \ln (1 - P_{s}) \right], \tag{4.6}
\]

for \( P_{s} = F(x_{s}^{\top} \beta) \), where \( q_{s} = 1 \) if \( y_{s} = 1 \), and 0 otherwise and where the function \( F \) is some strictly monotonic cumulative distribution. For logit regression, \( F \) takes the form

\[
F(x^{\top} \beta) = \frac{\exp(x^{\top} \beta)}{1 + \exp(x^{\top} \beta)},
\]

ensuring strict concavity of (4.6) w.r.t. \( \beta \), with first-order conditions

\[
\frac{\partial \ln L}{\partial \beta} = \sum_{s}(q_{s} - F(x_{s}^{\top} \beta))x_{s} = 0,
\]

where the term in brackets can be interpreted as an error in regression. For probit regression, \( F \) is the cumulative normal distribution and more demanding computationally as it is not available in closed form. The underlying choice function reads for this case of binary choice: \( c(x) = \text{sgn}(F(x^{\top} \beta) - \frac{1}{2}) \). Obviously, the maximum likelihood formulation (4.6) runs a greater danger of misspecification than SV-classification but it has the advantage that the function \( F \) admits an interpretation as a probability. Yet, the objective (4.6) might, for specifications of \( F \) other than logit, admit multiple stationary points. Furthermore, \( F \) has no estimated coefficients, as such estimation would be difficult because of possible multiplicity of stationary points and because the observations only indicate, for any \( x \)-value, whether it corresponds to an \( F \)-value above or below \( \frac{1}{2} \).

Since logit and probit are very widely used in applied econometrics and increasingly in social sciences at large, while binary classification is common in life sciences and engineering, we elaborate further on the difference between both maximum likelihood terms. Suppose that instead of maximizing the log-likelihood of the sample being as observed, like in logit/probit formulation (4.6), we seek to maximize the difference log-likelihood of as observed and the converse, for:

\[
\ln L = \frac{1}{S} \left[ \sum_{s=1}^{S} (\ln P_{s} - \ln (1 - P_{s})) + \sum_{s=0}^{S} (\ln (1 - P_{s}) - \ln P_{s}) \right], \tag{4.7a}
\]

\(^{14}\) In the GRCP-package of Appendix C by amending the program CLvBAG.GMS.
for the flexible functional form $\ln P_s - \ln(1 - P_s) = \sum_i w_i \psi_i(x_s) + \sum_j \phi_j(x_s)(\beta_j - \beta_j^*)$. This can be rewritten

$$\ln L = \frac{1}{S} \sum_s y_s \ln(\ln P_s - \ln(1 - P_s)).$$

(4.7b)

which amounts to minimizing the mean error $\xi_s$ calculated, in case of the eigenfunction form (as before the kernel form is obtained by replacing $\sum_i w_i \psi_i(x_s)$ with $\sum_r \alpha_r k(x_r, x_s)$), as:

$$\xi_s = \max(-y_s(\sum_i w_i \psi_i(x_s) + \sum_j \phi_j(x_s)(\beta_j - \beta_j^*), 0), (4.8)$$

and after introducing soft margin penalization and allowing for regularization leads us back to (4.3). As before, SV-classification based on (4.8) has the limitation that the estimated difference function is not a probability and, because of the non-parametric term, does not necessarily map on the unit interval. The important advantage is, obviously, that there is no need to postulate a logit or a probit form.

Next, noting that $\ln P_s - \ln(1 - P_s) \geq 0$ holds if and only if $P_s \geq \frac{1}{2}$, we consider the alternative formulation that comes even closer to logit/probit, and work with

$$\ln L = \frac{1}{S} \sum_s y_s \ln(\ln P_s - \ln(\frac{1}{2})),$$

(4.9a)

for $\ln P_s = F(\sum_i w_i \psi_i(x_s) + \sum_j \phi_j(x_s)(\beta_j - \beta_j^*))$, where $F$ is as before a strictly increasing distribution function. This leads to the amendment of the binary SV-classification (4.3) that can include this function $F$, specifically its inverse, so as to obtain:

$$\min_{\beta_j, \beta_j^*} \xi_s, \gamma_s \geq 0, \sigma \geq 0, w_i, \lambda \sum_i w_i^2 + \sum_j \tau_j (\beta_j + \beta_j^*) - \nu \sigma + \frac{1}{S} \sum_s g_s \xi_s,$$

subject to

$$y_s(\sum_i w_i \psi_i(x_s) + \sum_j \phi_j(x_s)(\beta_j - \beta_j^*) - F^{-1}(\frac{1}{2})) \geq \sigma - \xi_s$$

(4.9b)

which, since $\phi_1(x) = 1$ by assumption, leads to precisely the same coefficient estimates as in (4.3) and to the classification for (4.8), except that the intercept $(\beta_1 - \beta_1^*)$ is shifted by $F^{-1}(\frac{1}{2})$.

Furthermore, (4.9) makes it possible to keep $P_s$ on the unit interval, while the range of $f$ in (4.3) is unrestricted. However, (4.9) also shows that in this setting, irrespective of the specification of the function $f$, risk minimization with absolute value, quadratic value or any other convex form can never identify the shape of the function $F$.

This is not specific to logit-probit estimation. Any maximum likelihood method can at best postulate the functional form for the likelihood it has postulated. This is the basic motivation of distribution free methods (see Vapnik, 1998; and Györfi et al., 2002).
regression a form $F$ is to be postulated a priori and outcomes depend on it, in SV-regression the form can be postulated as well but the outcomes are insensitive to the choice, which seems preferable albeit that it seriously puts any interpretation of $F$ as a probability to question.

**Relation to Bayes problem**

The SV-classifier also has an interesting relationship to the underlying Bayes problem. For binary \{0,1\} valued $y$ and real-valued $x$, the conditional probability of $y = 1$ can be written:

$$\eta(x) = P(y = 1 | x) = E[y | x],$$

and the associated optimal decision that minimizes probability of error is:

$$g^*(x) = \begin{cases} 1 & \text{if } \eta(x) > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

As noted in Devroye et al. (1996, p. 20) the minimization of the sum of absolute values of

$$J(c) = E|\tilde{c}(x) - y|,$$

gives decision function $g^*$. Similarly, for vanishing regularization factor $\lambda$ and soft-margin penalizer $\nu$, the SV-classifier estimates the median of the conditional density, and, therefore, leads to (4.3), providing further justification for this specification.

**4.2.2 Single Class choice: quantile estimation**

The single-class problem can be used to select survey data from a larger set. It operates exactly as binary choice but with $y_s = 1$ all $s$. The soft margin penalization $\nu$ will no longer force separation, as there is no $y_s = -1$, but it will cause superfluous observations to be dropped, and clearly, the higher the regularization factor $\lambda$ relative to $\nu$, the lower the number of observations that will be kept. Consequently, the fit will be worse, and by the same token, $f(x) > 0$ will hold for fewer observations.

Restricting the number of observations to be dealt with, can in this way help making the SV-regression and classifications problems more tractable in size (but solving this problem for large data sets may in itself be demanding!). Usually, the “true” dependent variables, as opposed to the trivial $y_s = 1$, will have to be made part of the $x$-vector in this case.

**4.2.3 Multiple class choice: class wise versus joint estimation**

Multiple class choice can be conducted in two ways. The most common and straightforward way is to estimate a binary choice function for every class $h$ separately in a first stage, and apply the maximum rule (4.1) as selection device at a second stage. Since only one class has a positive value at every observation, the separately estimated functions will tend to generate the correct class selection at the second stage. Yet, the second stage is not part of the estimation. Furthermore, the functional form of the estimated functions should allow for positive and
negative values. It will usually be difficult to derive such a form directly from theoretical model of discrete choice and hence to impose an adequate specification for the parametric terms.

Alternatively, one may opt for joint estimation by SV-classification, starting from a theoretical model of discrete choice that is based on maximization of a (unobserved) payoff. For ease of exposition, and to provide a bridge to a common field of application, we introduce this approach by means of a standard micro-economic model of discrete choice by an individual consumer. This consumer can spend his given income \( m \) on one of \( H \) different indivisible objects, as well as on one divisible “other” good. Hence, he faces the budget constraint:

\[
\sum_h p_h c_h + p_o c_o \leq m,
\]

while
\[
\sum_h c_h = 1, \quad c_h \in \{0, 1\},
\]

where \( p_h, p_o \) are the given price of object \( h \) and the divisible good, respectively, and \( c_h, c_o \) are the associated quantities (\( c \) for consumption not to be confused with the \( c(x) \) of the choice function). We assume that \( p_h < m \) for all \( h \) (otherwise the class can be dropped) and postulate a utility function \( u : \mathbb{R}^{H+1} \to \mathbb{R}^+: u(c_1,\ldots,c_H,c_o) \), taken to be continuous, increasing. This defines the consumer model of discrete choice

\[
\max_{c_i \in \{0,1\}; h=1,\ldots,H} \max_{c_o \geq 0} u(c_1,\ldots,c_H,c_o) \quad \text{subject to} \quad \sum_h p_h c_h + p_o c_o \leq m, \quad \sum_h c_h = 1, \tag{4.10}
\]

which can be rewritten as:

\[
\max_{h=1,\ldots,H} u_h \left( \frac{m-p_h}{p_o} \right), \quad \text{subject to} \quad \sum_h p_h c_h + p_o c_o \leq m, \quad \sum_h c_h = 1, \tag{4.11}
\]

where \( u_h \left( \frac{m-p_h}{p_o} \right) = u(0,\ldots,0,1_h,0,\ldots,0,\frac{m-p_h}{p_o}) \), and \( 1_h \) denotes unity for the \( h \)-th element.

Now \( u_h \) are the functions to be estimated and given such functions we can use (4.1) as before. However, as utility \( u_h \) is not an indicator function (it is not positive when \( h \) is chosen and negative otherwise), we cannot include the information on the observed choice by pre-multiplication with \( y_s \) the way this was done in (4.3). To obtain an indicator, we may use for every class \( h \) the gap in utility of class \( h \) selected, and by assumption optimal, at observation \( s \) from class \( h \), and require non-negativity of this difference as an expression of optimality. Hence, there are now \((H-1)\) restrictions to be imposed at every observation. For a vector of independent variables with entries \( \frac{m-p_h}{p_o} \) for all \( h \), and possibly other determinants \( z \) of utility as well (leading to \( u(c_1,\ldots,c_H,c_o;z) \), this leads, for the case of given eigenfunctions to the SV-classification:
\[
\begin{align*}
&\min_{\beta_{hj}, \beta^*_hj, \xi_{hs} \geq 0, \sigma \geq 0, w_{hi}} \quad \frac{\lambda}{2} \sum_h \sum_i \omega^2_i + \sum_h \sum_j r_j (\beta_{hj} + \beta^*_hj) - v\sigma + \frac{1}{S} \sum_h \sum_s g_{hs} \xi_{hs} \\
&\text{subject to} \\
&f_{hs} = \sum_i w_{hi} \varphi_{hi}(x_s) + \sum_j \phi_{hj}(x_s)(\beta_{hj} - \beta^*_hj) \\
&f_{h,s} - f_{hs} \geq \sigma - \xi_{hs}, \quad h \neq h_s, \text{ for } h_s \mid y_{h,s} = 1.
\end{align*}
\]

and for the case with given kernel functions, to:

\[
\begin{align*}
&\min_{\alpha_{hr}, \beta_{hj}, \beta^*_hj, \xi_{hs} \geq 0, \sigma \geq 0} \quad \frac{\lambda}{2} \sum_h \sum_r \alpha_{hr} \sum_s k(x_r, x_s)\alpha_{hs} + \sum_h \sum_j r_j (\beta_{hj} + \beta^*_hj) \\
&\quad - v\sigma + \frac{1}{S} \sum_h \sum_s g_{hs} \xi_{hs} \\
&\text{subject to} \\
&f_{hs} = \sum_r \alpha_{hr} k(x_r, x_s) + \sum_j \phi_{hj}(x_s)(\beta_{hj} - \beta^*_hj) \\
&f_{h,s} - f_{hs} \geq \sigma - \xi_{hs}, \quad h \neq h_s, \text{ for } h_s \mid y_{h,s} = 1.
\end{align*}
\]

where $\alpha_{hr}$ is non-negative as before in (4.3), because we only impose lower bounds, but the pre-multiplication $y_{hs}$ in the regularization is omitted since there is no pre-multiplication in the constraint. We note that the non-negativity of the utility can be maintained by choosing a non-negative kernel function and by constraining the $\beta$-coefficients. Continuity readily follows by assuming continuity of $k$ and $\phi$. This also makes it easy to maintain increasingness for the parametric part. However, the non-parametric part function will in general not be increasing in $c$. Hence, in applications where this property matters (i.e. when prices and incomes are being varied), only the parametric part can be used, and the non-parametric part is to be dealt with as a fixed effect at most.

We also remark that even for the case with two classes the estimation procedure is different from the common SV-classification (4.3), because we estimate utility functions rather than utility gap functions.

To sum up, in multiple class problems, class wise estimation is less demanding in dimensionality but simultaneous estimation is needed if theoretical properties of the underlying model have to be maintained.

### 4.2.4 Multiple class choice: multistage form

So far, we considered the choice among a finite number of options that are simultaneously available. Economic applications often relate to multistage situations say, to represent decision trees. In their most general (least structured) form, multistage problems can be expressed as multiclass problems, with every end-node of the tree represented by a separate class $h$. In more structured situations, the decision branches can be analyzed and estimated separately. The question is then how to incorporate conditioning, when deemed relevant, for example the property that “of the individuals choosing to renovate their house, some decide to buy a particular brand of heater”, or in a multi-agent context, “of the individuals eligible for subsidy on renovation, some choose a brand of heater”.
The sequential approach to conditioning proceeds with separate estimations for every stage. One possibility is to include the observed first-stage choice (living in this particular city) as an independent, binary (dummy) variable in the second stage choice that essentially allows the parametric term to differ in some way for this city, and possibly enters the kernel function as well. More common in econometrics is “instrumentalization”, which incorporates either the predicted discrete choice, the predicted \( f \)-value or the predicted parametric value within the second stage equation. Instrumentalization has the drawback that it attributes higher priority to risk minimization of early stages.

The simultaneous approach would impose risk minimization on the full problem while applying instrumentalization. In terms of SV-classification (4.3) this creates the following difficulties. As the predicted discrete choice generates a non-convexity, only the option of using the predicted \( f \)-value and the predicted parametric value are available. Furthermore, these predicted values cannot enter as elements of the next-stage \( x \)-vector within a single SV-program as this would cause non-convexity, even in the purely linear, parametric case with \( f(x) = x^T \beta \). On top of that, the chain of multistage choices would naturally lead to product forms of functions \( f \) that would also cause non-convexity. Yet, various options are available to represent the interdependence among the stages within a single program. For example, if in a two-stage situation the realization \( y_s^1 = 1 \) in stage 1 is conducive to \( y_s^2 = 1 \) in stage 2, the constraints of simultaneous SV-classification can be written:

\[
y_s^1 f^1(x_1) \geq \sigma - \bar{\xi}_s^1, \text{ for stage 1}
\]

\[
y_s^2 (f^2(x_1, x_2) + f^1(x_1)) \geq \sigma - \bar{\xi}_s^2, \text{ for stage 2, or, alternatively,}
\]

\[
y_s^2 (f^2(x_1, x_2) + q^1_s f^1(x_1)) \geq \sigma - \bar{\xi}_s^2.
\]

Many other forms with inequality constraints involving several functions can be composed but since the structure of such constraints is highly dependent on the problem at hand, we do not include this simultaneous multistage classification as a pre-specified option in the GAMS-program of Appendix C.\(^{16}\)

### 4.3 Classification by kernel smoothing (mollifier)

As in Section 3.4, we take the kernel to be a density when considering kernel smoothing (without accounting for district data). This leads, by averaging over binary observations, and disregarding the window size for notational convenience, to the following nearest-neighbor type classifier:

\[
c(x) = \text{sgn}(\sum_s y_s k(x_s, x)) = \text{sgn}(\sum_{s|y_s=1} k(x_s, x) - \sum_{s|y_s=-1} k(x_s, x)).
\] (4.9)

This indicates that for kernel smoothing, the decision is driven by the difference in likelihood, i.e. essentially obeys a maximum likelihood rule, or kernel-weighted majority vote, for the density specified. Clearly, the likelihood of \( y_s = 1 \) prevailing at \( x \) can in this way be obtained as \( \ell^S_+(x) = \frac{1}{S} \sum_{s|y_s=1} k(x_s, x) \), and the probability as \( L^S_+(x) = \frac{\ell^S_+(x)}{\ell^S(x)} \) for

\(^{16}\) Instead, the user may modify GAMS-program CLvBAG.GMS to create the appropriate setting.
\[ \ell^S(x) = \frac{1}{S} \sum_s k(x_s, x). \] As for SV-regression we use kernel smoothing to estimate likelihoods of errors, in Section 4.7 below.

### 4.4 Nearest neighbor

As mentioned above, the nearest neighbor classifier chooses the class \( y \), indicated by majority vote among \( N^S \) nearest neighbors in the survey with data set \((x_s, y_s)\), e.g. with \( N^S = (S)^a \), \( a \in (0,1) \). Hence,

\[
h(x) = \max_y \sum_n w_{ys_n}(x),
\]

where \( w_{ys_n}(x) = 1 \) if \( y_{s_n}(x) = y \) and 0 otherwise, is the vote for \( y \) by the \( a \)-th nearest neighbor at \( x \), defined by \( s_n(x) = \arg \min_{s \in S_n(x)} \| x_s - x \|^2_{\Omega^{-1}}, \) for candidate set \( S_n(x) = \{ s \in \{1,...,S\} \mid s \neq s_n(x), n'=1,...,n-1 \}, \ n = 1,...,N^S; \ s_0(x) = 0, \) and where \( \Omega \) denotes the given covariance among \( x \) (usually the inverse of the variance of separate elements \( x_{is} \) in the survey). The technique cannot account for district data but it can be used in large-scale applications, for example of quantile estimation to limit the size of a data set.

### 4.5 District data in SV-classifier

At district level, observations may be available on the fraction \( N^+_d \) of positive choices (e.g. votes) \( y_s = 1, s \in S \) for every class \( h \). However, because of the discrete-valuedness of the choice function, straight imposition of district information either as data or as constants, is not a practical option, as it would lead to a very large mixed integer program. Alternatively, approximating the choice function by some continuous step function would create a non-convexity and hence seems impractical as well. Therefore, we consider a less precise approach based on two remarks.

First, at district level the discrete choice can, for heterogeneous populations, be approximated by a continuous choice, because the switches at individual level have little effect on the total since they occur for different parameter values \((\alpha, \beta)\). Second, it may acceptable to assume that the mean value of the estimated function over the sample applies rather closely to the population as a whole.

The information at district level can, before approximation, be represented by specifying the fraction of positive votes:

\[
N^+_d = \frac{\sum_{s \in G} \delta^d_s \kappa^+ (f_s)}{\sum_{s \in G} \delta^d_s}
\]

where \( \kappa^+ (f) = 1 \) if \( f > 0 \) and 0 otherwise, and
\[ f_s = \sum_{r \in G^o} k(x_s, x_r) \alpha_r + \sum_j \phi_j(x_s)(\beta_j - \beta_j^*) \]. \tag{4.11b}

To obtain a linear district constraint, we assume that \( F_d = \frac{\sum_{s \in G} \delta_s f_s}{\sum_{s \in G} \delta_s} \), the aggregate value of the function taken over the full census, is approximately equal to the weighted sum of function value in the survey, for positive and negative votes

\[ \hat{F}_d \approx N_d^+ f_d^+ + N_d^- f_d^- , \tag{4.14} \]

for \( N_d^- = 1 - N_d^+ \), district aggregate \( \hat{F}_d = \sum_{s \in G} f_s \) and sample means:

\[ f_d^+ = \frac{1}{S_d^+} \sum_{s \in G^o | y_s = 1} \delta_s d f_s , \tag{4.15a} \]

\[ f_d^- = \frac{1}{S_d^-} \sum_{s \in G^o | y_s = -1} \delta_s d f_s . \tag{4.15b} \]

where \( S_d^+ = \sum_{s \in G^o | y_s = 1} \delta_s d, S_d^- = \sum_{s \in G^o | y_s = -1} \delta_s d \), both taken to be positive. Then, we can for given constant \( \gamma_d \) express the constraint by district as:

\[ -\gamma_d \leq V_d \leq \gamma_d , \tag{4.16a} \]

\[ V_d = N_d^+ f_d^+ + N_d^- f_d^- - \hat{F}_d , \tag{4.16b} \]

where we allow for positive \( \gamma_d \) to express that (4.14) is an approximation. This makes it possible to formulate the program with aggregate district constraints that are linear in the choice variables \((\alpha, \beta)\). The program below treats the district data as constraints all expressed in primal form with kernel expression, as in (4.5):

\[
\begin{align*}
\min_{\alpha_s, \beta_j, \beta_j^*, \xi_s, \sigma \geq 0, f_d^+, f_d^-, G_s, y_s} & \quad \lambda \sum_{r \in G^o} \alpha_r y_r \sum_{s \in G^o} k(x_r, x_s) y_s \alpha_s \\
& \quad + \sum_j \tau_j (\beta_j + \beta_j^*) - v \sigma + \frac{1}{S} \sum_{s \in G^o} g_s \xi_s \\
\text{subject to} & \quad y_s f_s \geq \sigma - \xi_s , \quad s \in G^o \\
& \quad f_s = \sum_{r \in G^o} k(x_s, x_r) \alpha_r + \sum_j \phi_j(x_s)(\beta_j - \beta_j^*) \\
& \quad V_d = N_d^+ f_d^+ + N_d^- f_d^- - \sum_{s \in G} \delta_s d f_s \\
& \quad f_d^+ = \frac{1}{S_d^+} \sum_{s \in G^o | y_s = 1} \delta_s d f_s \\
& \quad f_d^- = \frac{1}{S_d^-} \sum_{s \in G^o | y_s = -1} \delta_s d f_s \\
& \quad -\gamma_d \leq V_d \leq \gamma_d ,
\end{align*}
\]
Once the estimation has been conducted, it is possible to check the accuracy of the approximation by evaluating the choice or decision function $c(x)$ over the census, and possibly adjusting the coefficients $\gamma_d$ for a new round of estimation.

While specification (4.17) refers to the treatment of district data as given, which we may recall has the advantage that it supports an evaluation of probability bounds as in section 3.6, collection of terms also would allow for a treatment of the district information as regular data, since

$$V_d = \frac{N_d^+}{S_d^+} \sum_{s \in G^+ | y_s = 1} \delta_s^d f_s + \frac{N_d^-}{S_d^-} \sum_{s \in G^+ | y_s = -1} \delta_s^d f_s - \frac{\sum_{s \in G} \delta_s^d f_s}{\sum_{s \in G} \delta_s^d}$$

will, after substituting the expression (4.11b), be linear in the coefficient $(\alpha, \beta)$.

A straightforward modification of (4.17) writes eigenfunctions in the expression for $f_s$ in (4.17), and replacing the quadratic term in $\alpha$ in the objective by a quadratic term in $w$, as in (4.3). Similarly, joint estimation can be implemented as in (4.12), (4.13) with all constraints of (4.15) and (4.16) introduced for all classes within the same program.

Finally, we note that supposing that the district constraints are correctly specified, $f(x)$ converges under the same conditions as for continuous data, essentially because the function $f$ and hence the associated loss function is the same.

### 4.6 Ranking while accounting for district data

In between discrete choice and limited dependent but real-valued variables is the problem with data on ranking (well known in econometrics as ordered-logit or ordered probit). There is a single function, whose range is subdivided into intervals, and a single estimation is conducted for all classes simultaneously. The choice of interval defines the classification. Hence, as there are both lower and upper bounds the parameters $\alpha$ should be left unconstrained in sign. The dependent variable is integer-valued but refers to a ranking (the function value in class $h$ is always less than class $h + 1$) according to fixed thresholds separating the classes to be denoted by lower bounds $\bar{f}_h$, $h = 1, \ldots, H + 1$ and enclose all values.

To reflect the district constraints we suppose that the district observations refer to the middle of the relevant interval, i.e. that the aggregate gap can be written:

$$V_d = \sum_h \frac{N_{hd}}{2} (\bar{f}_{h+1} - \bar{f}_h) - \frac{\sum_{s \in G} \delta_s^d f_s}{\sum_{s \in G} \delta_s^d}$$

The thresholds $\bar{f}_h$ are either given or to be estimated. In the latter case, the program reads:

$$\min_{\beta_j, \beta_j^*, \xi_s, \sigma \geq 0} \lambda_s \sum_{r \in G^o} \alpha_r \sum_{s \in G^o} k(x_r, x_s) \alpha_s + \sum_j \tau_j (\beta_j + \beta_j^*) - \nu \sigma + \frac{1}{S} \sum_{s \in G^o} g_s \xi_s$$

---

17 See Appendix B for the implementation. We are not aware of such formulations in kernel learning.
subject to

\[ f_s = \sum_{r \in G^o} k(x_s, x_r) \alpha_r + \sum_{j} \phi_j(x_s)(\beta_j - \bar{\beta}_j^s), \ s \in G^o \]

\[ \bar{f}_{h+1} - f_s \geq \xi_s - \sigma, \quad \text{if } y_s = h, \ s \in G^o, \]

\[ f_s - \bar{f}_h \geq \xi_s - \sigma, \quad \text{if } y_s = h, \ s \in G^o. \]

\[ V_d = \sum_h \frac{N_{hd}}{2} (\bar{f}_h + \bar{f}_{h+1}) - \frac{\sum_{s \in G} \delta_s^d f_s}{\sum_{s \in G} \delta_s^d} \]

\[ -\gamma_d \leq V_d \leq \gamma_d, \]

Once estimated, the function \( f \) defines a classification. In fact, from the estimated function we can obtain the continuous functions:

\[ f_h(x) = \min(f(x) - \bar{f}_h, \bar{f}_{h+1} - f(x)), \]

that through their sign define a binary (member/non-member) decision w.r.t. every class \( h \). If the thresholds are known, they disappear from the list of choice variables, and the district constraints can use an exogenously given mean of every class. As before, eigenfunctions are readily incorporated to replace kernel functions.

### 4.8 Probability and likelihood bounds

The approach to calculation of probability and likelihood bounds is a direct extension of the discussion in Section 3.6 above that will, therefore, not be repeated here. Specifically, in analogy to the calculation of the likelihood of a within-range deviation in (3.23), we can assume the kernel to be a density, as in the kernel smoothing of 4.2 above, and use this to evaluate the likelihood of the binary kernel learning classifier agreeing with kernel smoothing as:

\[ L_k(x) = \frac{1}{S} \sum_{y_s = 1} k(x_s, x) \text{ if } c(x) = 1 \text{ and } \frac{1}{S} \sum_{y_s = 0} k(x_s, x) \text{ otherwise} \]

Obviously, the calculation is only indicative, since the kernel will usually fail to coincide with the true density. Nonetheless,

(a) a high value means that the kernel learning projection is (i) well based on empirical evidence and (ii) agrees with the projection by kernel smoothing.

(b) a low value means that the kernel learning projection is either (i) not well based on empirical evidence; or (ii) disagrees with the projection by kernel smoothing.

The associated conditional probabilities can be obtained through division by the likelihood \( \frac{1}{S} \sum_s k(x_s, x) \). For the ranking problem, the likelihood density from kernel smoothing can be obtained as:

\[ L_k^S(x) = \frac{1}{S} \sum_{s \in G^o} k(x_s, x), \]
where $G_h^o$ refers to the set of observations in class $h$. 
5. Micro-simulation and prediction of district aggregates

For the continuous dependent variables of section 3, prediction at district level mainly serves as a tool for micro-simulation of the effect of changes in some of the independent variables. Indeed, as long as the independent variables are kept at the census-values used in estimation, district level prediction amounts to aggregation and hence to reproducing the district constraints themselves. Clearly, the sensitivity of the district estimate should not be assessed by aggregation of variances from bagging while assuming independence across underlying \(x\)-values. Since the census is large the resulting mean would always have virtually zero standard deviation. Taking the mean standard deviation would give a more conservative estimate. Yet, the imposition of the district constraint itself as an (almost) equality necessarily makes it difficult to retrieve information on stability of prediction at that level. This is a topic for further research.

When dependent variables are discrete, prediction at district level is less straightforward, especially for non- and semi-parametric specifications. Parametric methods of binary choice estimation such as probit or logit usually limit the range of the function \(f\) on the unit interval and under prediction interpret this function as a probability of occurrence of the event at the level of the individual. This makes it possible to evaluate the probability of positive vote by an individual at the mean of the district, and also to evaluate the mean probability of a positive vote in the district or within any pre-specified range of \(x\).

Under the semiparametric formulation of section 4, the interpretation at the level of the individual would seem less natural. Since all coefficients \(\alpha\) of the binary choice are assured to be nonnegative, observations \(y\) are binary, and coefficients \(\beta\) could be constrained in sign, it is for nonnegative kernel functions relatively straightforward to limit the range of \(f\) for SV-classification as well. However, especially for out-of-sample predictions, the range extends beyond the \([-1,1]\) interval of the observations rather widely, and this would make the interpretation as a probability of the projection onto \([0,1]\) questionable. Therefore, in Appendix C we do not evaluate a probability of occurrence at the level of the individual and in the case of binary choice we only allow for calculation of the probability of occurrence at district level, evaluated as the share of positive votes in the total, and for \(x\) within a user-specified district wise domain that could lie outside the sample (as one way to enforce a simulation scenario, the other being an exogenously specified, district wise shifter on \(x\)).

The cases of multiple choice and ranking can be reported on in a similar way for every class separately, as the probability of a winning (as opposed to a mere positive) vote. However, they require selecting from the vast amount of polling information as to the robustness of the choice, which can be reported on as well. At the level of the individual, we allow for (i) by class computation and mapping of \(f\)-values with their standard deviations and likelihood and probability calculations, as for continuous variables; and (ii) overall computations and maps indicating for every individual the label of the winning class, the label of the runner up, the \(f\)-value of the winner, and the edge of the winner over the runner up (for the total prediction and for the parametric part only). At the level of the district, available options are to compute (i) by class, the percentage of individuals in the district for whom this class is the winner, and for whom it is the runner-up, within a pre-specified domain of \(x\), and (ii) overall, the winning class and the mean winner’s edge over the runner-up, for the same domain.
To sum up, rule-based gridding has the advantage that it can account for a priori restrictions, also at points where no data are available. Regression and classification techniques seem less ad hoc and are, more importantly, open to explicit empirical testing. Yet, as it often proves difficult to incorporate all a priori restrictions in a common regression function, the most powerful mode may require taking the approach one step further by operating on selected parameters of a well-structured model that meets all a priori restrictions, rather than on a regression model, adjusting these parameters so as to optimize the fit. However, as this model is necessarily application-specific, the practicability of this intermediate path can only be established on a case-by-case basis.

Further work

Priorities for further work include the inclusion of practical rules to establish proper values of regularization parameters $\lambda$, $\tau$, to test the sensitivity of the kernel function to variation in window size, and more generally to formulate tractable kernel functions.
Appendix: GRCP, a GAMS-controlled package for Gridding, Regression, Classification and Plotting

A. Gridding and plotting of maps

The gridding software operates from a GAMS-platform, which is used to generate ASCII input files for Fortran programs that perform the gridding and generate Excel-files and SAS-command files for automatic preparation of geographical maps. GAMS is only a used as a job control language to generate commands in user-written program. Actual calculations are done by the Fortran executables (GRIDDIST for gridding and plotting of grid files, and SPLITN for plotting of district files at county, province or regional level), which in turn write SAS-programs and Excel files for plotting of maps. This Appendix documents the major steps of such a program. New users are advised to prepare such a program by modifying an existing application. Allowing for control via a GAMS-program has the advantage over a pure screen driven Windows-application that the user is more flexible in preparing the data and that the program offers a transparent documentation of the full application.

A DOS-script is used to run the GAMS/Fortran/SAS sequence. In addition, there is a Fortran program that makes the gridded files generated by GRIDDIST available as aggregates at district level in a GAMS-readable file (use CALL AGGREGATE.EXE <filename> in this DOS-script to generate this file). This file can be accessed by a subsequent GAMS-program via an INCLUDE-statement.

SASBASE.SAS created in directory ..\WKRUN is an independent SAS-job (invoking application-specific macro’s) that is readily modified to perform other preprogrammed SAS-functions, such as plotting of histograms or curves.

Finally, the data generated for plotting are stored in WKRUN on Excel files and can also be accessed directly in Excel for further processing.

Gridding operates at census or grid level either by scaling a reference variable to the extent it exceeds a given threshold, and possibly within bounds, or by conducting arithmetic operations on it. Central guideline in all operations is that "a zero reference variable keeps the outcomes at zero at that site, whatever the bounds". Hence, the reference variable solely controls the non-zero positions. The scaling is by administrative district and such that wherever on the grid there is a nonzero reference variable exceeding the specified threshold in this district, the values on the grid are adjusted according to some rule, until they sum to the district total.

A.1 Gridding rules

The rules are defined in GAMS as a SET:

```
SET RULE ' Rule of operation ' /
* scaling to total
1 ' scaling no bounds
```

Here $b^+$ refers to an additive shift in the distribution, $b^*$ to a multiplicative shift. “Average as bound” means that the district average (over sites with positive reference variable) is used to define the bound, which is obtained after multiplication of this average by a factor.

The reference vector $\hat{x}$ is pre-processed to put values below the threshold equal to zero:

$$\hat{x}_s = x_s \text{ if } x_s \geq \tau x_s^0 \text{ and } 0 \text{ otherwise},$$

for a scalar threshold factor $\tau$ and a threshold base vector $x$.

A zero district variable imposes zero values on all grid cells in the district, whatever the rule. This is to allow for districtwise calculations.

For rules 1-6 further adjustment is done to meet the given district totals, within bounds. Under rules 1-5, the operation is on the reference variable scaled to meet district total, as in (2.1).

$$y_s = \min(\max(a_s + b_s \hat{x}_s, \bar{y}_s), \bar{y}_s),$$

$$\sum_s \delta^d_s y_s = Y^d.$$

If the bounds cannot be met processing is as follows: (i) if the district total falls below the lower bound, the lower bound is reset to zero and scaling proceeds; (ii) if the district total exceeds the upper bounds, the upper bound prevails and all values in the district are set equal to it.

Under Rule 6, priority allocation, program (2.4) is solved. Finally, rules 7-10 do not apply any scaling and operate directly on the reference variable and the bounds. The bounds are obtained by multiplication of a reference vector with a constant:

$$\bar{y}_s = \tau x_s, \bar{y}_s = \tau x_s$$

---

For ease of exposition, we refer to all symbols in the way they are in sections 2-3 of the paper, even though the GAMS program only accepts ANSI-symbols.
Hence, the procedure needs three scalars \( \hat{\tau}, \underline{\xi}, \overline{\tau} \) for the threshold and the two bounds, as well as four pointers: one to the rule, and three to the variables (grid-vectors) the scalars should operate on.

A zero-level reference variable automatically induces zero-level bounds. For rules 1-6 the bounds should be consistent and such that their district wise sums over sites with positive reference variable should define intervals that comprise the district totals. For rules 7-10, district totals are not used and the bounds are processed whatever their values, i.e. the rules define mere arithmetic operations.

**A.2 Grid files**

Next, we turn to the establishment of reference variables stored as grid files (rasp. census files), defined in the set REFER, which obviously differs by application:

```gams
SET REFER 'Reference variables' /
F0 ' 1 zero',
F1 ' 2 one',
RURP ' 3 rural population',
URBP ' 4 urban population',
TOTP ' 5 total population',
POTY ' 6 potential yield',
TPO2 ' 7 total population updated',
PRKR ' 8 pork',
WRK1 ' 9 file 1 for temporary use',
WRK2 '10 file 2 for temporary use' /

SET REFERINI(REFER) 'Reference variables available initially' /
RURP,
URBP /

SET REFERNAM(REFER) 'Name of files of reference variables' /
RURP,
URBP,
TOTP,
POTY,
TPO2,
PRKR,
WRK1,
WRK2 /

SET REFERINI(REFERNAM)'Reference variables available initially' /
RURP,
URBP
```

This list should be comprehensive, comprising all grid files used as input in subsequent GAMS-programs for gridding. Files generated by the program but used only as input for plotting can be omitted. The file labels (excluding the suffix .grd that will be appended later on) should differ in their first four characters (used for file name) and may be up to 20 characters long. As subsets of REFER, we also define the set REFERNAM that excludes the reference variables F0 and F1, which are known a priori, and REFERINI with the files that are available at the beginning of the procedure.
**A.3 Gridding commands: OUTRULE and OUTASC2**

Commands are to be generated as a sequence of BATINCLUDE-statements for OUTRULE, completed by a single BATINCLUDE for OUTASC2. OUTRULE commands define the order of processing and may follow a sequence different from the order of the selection referred to in the correspondence of the OUTASC2-call. Any file generated by one OUTRULE command may overwrite input files of the command and will, if duly stored via the correspondence, be available for subsequent OUTRULE commands. Hence, the program can be written as if all commands were executed as and when they are given. Subroutine-calls require the following parameters.

**Subroutine OUTRULE:**
1. Element name (short label in index set)
2. Index set (with the long labels)
3. Order vector of index set
4. Selection from index set
5. Rule
6. Reference file (rank in REFER)
7. Threshold base
8. Factor on threshold base
9. Lower bounds (rank in REFER)
10. Factor on lower bounds
11. Upper bounds (rank in REFER)
12. Factor on upper bounds

**Subroutine OUTASC2:**
1. Matrix of data
2. Geo-index
3. Selection from index set
4. Index set (with the long labels)
5. Number of positions per variable on output
6. Number of decimals
7. Number of levels (colors/classes) requested for plot
8. Order vector of index set
9. Mapping from index set to file

**A.4 Example**

Here follows an example of such a sequence.
- Define index sets

**SET**
- full index set with labels

```
KPOP 'Population '
/
TOTPOP 'total population '
UPPOP 'urbanlike population '
LOPOP 'rurallike population '
OTHPOP 'other population '
/```
• selection
KFPOP(KPOP) 'Population 2'
/ TOTPOP
UPPOP
LOPOP
/

• Initialize population

PARAMETER
POPT(C,KPOP) "Population by county C"
;

• Here we assume that POPCN has been read in earlier

POPT(C,'TOTPOP') = POPCN(C);

• Store the ordering of full index set

PARAMETER ORDPOP(KPOP);
ORDPOP(KPOP) = ORD(KPOP);

• Define the correspondence that links the GAMS-variables and the grid-files

SET POPMAP(KPOP,REFER);

• Specify whether and if so how the calculations are to be stored on grid files (no storage = NO):

POPMAP(KPOP,REFER) = NO;
POPMAP('TOTPOP','TPO2') = YES;
POPMAP('UPPOP','WRK1') = YES;
POPMAP('LOPOP','WRK2') = YES;

• Enter the OUTRULE commands (here $... is used as shorthand for $BATICCLUDE ..\WKRUN\ in the actual GAMS-program)

* Make TPO2 available by scaling the population to district totals
$_OUTRULE.gms "TOTPOP" KFPOP ORDPOP KPOP 1 3 1 0 1 0 1 0.

* Select (rule 9) upper part via threshold of 100000 people
* Per gridcell. Store as 9
$_OUTRULE.gms "UPPOP" KPOP ORDPOP KFPOP 9 7 2 .05 1 0 1 0.

* Obtain lower segment (rule 9) subtracting file 9 from 7.
* Store as 10
$_OUTRULE.gms "LOPOP" KPOP ORDPOP KFPOP 9 7 1 0 9 1 1 0.

* Go
$_OUTASC2.gms POPT C KFPOP KPOP 14 5 NLEV ORDPOP POPMAP

This complete a gridding operation.
A.5 Plot map of district totals

Besides gridding and plotting of grid maps as controlled via OUTRULE and OUTASC2, the GAMS-jobs can also invoke a subroutine SASTAB2E to produce maps for every administrative level (regional, provincial or county). The example below illustrates this for a map of the commodities in the subset \(KFDIST(K)\) of rural consumption \(RUCONS(R,KFDIST)\), assuming that the subset has been defined, that descriptive labels are available for the parent index set \(K\) and that the matrix with the data is available.

- Before the actual commands are given, files must be defined as well as some basic parameters

FILE FILSAS /sasbase2.sas/;
FILSAS.PW=1000; FILSAS.PC=2;
PUT FILSAS / "#";

FILE FILREC /C:\CHINAPART\WKRUN\sasrecl.txt/;
FILREC.PW=100; FILREC.PC=2;

PARAMETERS
NLEV 'number of levels'
NLEVX 'maximum number of levels'
NLOC 'number of sites'
NRTAB 'number of tables'
NVAR 'number of variables'
NVARX 'maximum number of variables'

NRTAB = 0;
NVARX = 0;
NLEVX = 0;

- Next, the plotting routing can be called. Here we give an example for consumption:

* RUCONS matrix of data
* R GAMS geo-index (county/province/region index R,PV,CN or aliases thereof)
* KFDIST (commodity) selection from matrix
* K parent of KFDIST with the long labels
* 'OTHER XX' checking of data availability in index TABSEL
* XXX not used
* 4 number of levels (colors/classes) requested for plot
* R identifier for SAS should be R, PV or CN

$... SASTAB2E.GMS RUCONS R KFDIST K 'OTHER_XX' XXX 4 R

The names of the plot files (.gif) and the data in Excel (.csv) will now end with \(r, p\) or \(c\) depending on whether they relate to \(R\), \(PV\) or \(CN\). Grid plots are obtained via the OUTASC2-commands (e.g. rule 9 with zero change on reference variable). The file SASBASE2.SAS is input to the Fortran program SPLITN.EXE that generates the actual SAS-job run as SAS SASTRAN2.SAS using the Excel input files (.csv extension) stored in the subdirectory WKRUN.
B. Regression estimation and plotting of maps and functions

The Regdat-facility is a GAMS-controlled regression and mapping software, that combines district, with survey, census, and map information. The user writes a short GAMS-program that initiates the following tasks:

1. Preparation of data files for regression (Fortran)
2. Regression (GAMS)
3. Retrieve predicted values on survey data
4. Mapping out of regression function over census and preparation of SAS-files for plotting of maps (Fortran)
5. Aggregation of individual or grid data to district (or higher) level to produce district map
6. Running SAS-programs to produce grid maps.
7. Preparing 2D and 3D-graphs of functions

The DOS-script Regdat.bat is used to execute these tasks. A control list in GAMS specifies the sequence of tasks to be conducted.

B.1 Specification of tasks

```
SET TASKS    'Possible tasks'
    / TASK1    'prepare matrices for estimation '
    TASK2    'conduct estimation '
    TASK3    'predict function value on survey data '
    TASK4    'predict function value on census '
    TASK5    'produce district map '
    TASK6    'produce grid map '
    TASK7    'produce 2-3D function plot '
    /;
```

Tasks\(^{19}\) are activated by defining appropriate (uninterrupted sequences) as subset, for instance:

```
SET DOTASKS (TASKS)    'Tasks to be executed '
    / TASK1
    TASK2
    TASK3
    /;
```

Task 3 generates a file with the survey data set in GAMS-readable format, extended with the full prediction and the parametric prediction, for further tabulation in GAMS and for possible use as instrumented variables in subsequent regressions. Similarly for Task 4. Further controls on tasks are specified in section B.6 below.

\(^{19}\) A gridmap can only be produced if the census file is geo-referenced.
B.2 Options for regression

The user can choose between various regression techniques:

```plaintext
SET RGTECH   'Regression techniques ' /  
    EIGCON     'SV-eigenfunctions '  
    KERCON     'SV-kernel functions'  
    OLS        'OLS '  
    MOLL       'Mollifier '  
/;

SET DORGTECH 'Choose one technique ' /  
    KERCON     'SV-kercons'  
/;
```

where the Support vector (SV)-regressions are semiparametric as in section 3, with the $\beta$-coefficients associated to the parametric terms. Specifically,

- **EIGCON** primal Quadratic Program for SV-regression, for given eigenfunctions, district data fixed
- **KERCON** primal Quadratic Program for SV regression, radial basis kernel, district data fixed
- **OLS** ordinary least squares
- **MOLL** mollifier or Nadaraya-Watson estimation or kernel smoothing

As the mollifier is non-parametric, there is no need for parameter estimation, and all operations are conducted at the stage of prediction. District data are only processed for SV-regression and OLS.

```plaintext
SET SCALP    'Possible scales for aggregation and plotting ' /  
    REGP   'regions '  
    PROVP   'provinces'  
    COUNP   'counties '  
/;

SET DOSCALP(SCALP) 'Scale to be plotted ' /  
    COUNP   'counties '  
/;
```

B.3 Specification of grid and census files

Grid maps or census data are listed in the set REFER as in appendix A above but since in regression these files are exclusively used to store the independent variables for the regression on district means, we only need one list.

```plaintext
REFER    'Reference variables ' /  
    F0        ' 1 zero '  
/;
```
The file labels (excluding the suffix .grd that will be appended later on) should differ in their first four characters (used for file name) and may be up to 20 characters long.

### B.4 Selection of independent variables

Independent variables are listed in the index set $KX$.

```plaintext
SET KX 'independent variable '
/
KXX1
KXX2
KXX3
KXX4
/;
```

The matrix $KXX$ associates a consolidated list of independent variables for all classes to files in REFER. The first column entry always has to refer to the scaling variable $\delta_h^b$:

```plaintext
TABLE KXX(function,KX) "pointers of independent variables to files "

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>KXX1L</th>
<th>KXX2L</th>
<th>KXX3L</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIZE</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>PORK</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
```

The intercepts of functions should not appear in the first column and are recognized as references to the file $F1$ in the set REFER.

When applicable, the eigenfunctions use independent variables selected from the list $KX$ for every dependent variable separately. These variables enter functional forms to be specified in the next section:

```plaintext
TABLE KIX(KFDIST,KX) "select x for eigenfunctions "

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>KXX1L</th>
<th>KXX2L</th>
<th>KXX3L</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIZE</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>PORK</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
```

Here a zero entry means non-selection, and a positive entry establishes the number of terms in a function expansion (e.g., a polynom) in which the variable should enter. Similarly, for the parametric terms the independent variables are selected as:
TABLE KJX(DEFINE,KX) "select x for parametric terms"

<table>
<thead>
<tr>
<th></th>
<th>KXX1L</th>
<th>KXX2L</th>
<th>KXX3L</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIZE</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PORK</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Kernel functions will contain as arguments all independent variables except the scaling variable and the intercept.

**B.5 Control parameters for classification: district data, bagging, windowsize, functional form**

The control parameters for classification are defined in the set CLASSCONT and the table CONTVAL:

```plaintext
SET REGCONT ' control parameters for regression' /
  BAGNR ' number of bags' '1'
  BAGREP ' bag sampling with replacement (1=yes/0=no)' '0'
  DISTCON ' district constraints (1=yes/0=no)' '1'
  FTYPN ' function type non-parametric part 1=polynom, 2=1 in logs' '1'
  FTYPP ' function type parametric part 1=polynom, 2=1 in logs' '1'
  LDWEIGHT ' likelihood density weight error survey data (1=yes/0=no)' '0'
  NUFACSC ' factor on soft margin penalization' '1.'
  RGLFACSV ' factor on regularization constant SV-regression' '1.'
  WINDFACR ' factor on windowsize kernel used in regression' '1.'
  WINDFACL ' factor on windowsize kernel used in likelihood calculation' '1.'
  YMAP ' mapping on y: 1=identity 2=logarithm 3=inverse logistic' '1.'
/

TABLE CONTVAL(REGCONT,INIT) ' control parameters for classification'

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td></td>
</tr>
<tr>
<td>BAGNR</td>
<td>1</td>
</tr>
<tr>
<td>BAGREP</td>
<td>0</td>
</tr>
<tr>
<td>DISTCON</td>
<td>1</td>
</tr>
<tr>
<td>FTYPN</td>
<td>1</td>
</tr>
<tr>
<td>FTYPP</td>
<td>1</td>
</tr>
<tr>
<td>LDWEIGHT</td>
<td>0</td>
</tr>
<tr>
<td>NUFACSC</td>
<td>1.</td>
</tr>
<tr>
<td>RGLFACSV</td>
<td>1.</td>
</tr>
<tr>
<td>WINDFACR</td>
<td>1.</td>
</tr>
<tr>
<td>WINDFACL</td>
<td>1.</td>
</tr>
<tr>
<td>YMAP</td>
<td>1</td>
</tr>
</tbody>
</table>
;

Most of these parameters are self-explanatory.

**Bagging.** To assess the stability of estimates, bagging can be conducted whereby the survey data set is subdivided into an equal number (BAGNR) of subsets for which estimation is conducted separately, yielding subset-specific predictions. The bags can be built (a) by straight subdivision...
of the sample into separate parts (BAGREP=0); or (b) by sampling with replacement from the full data set (BAGREP=1). The prediction used is the mean over these predictions, and when bagging is active a standard deviation can be computed as well.

**Function type.** The parametric part and the nonparametric part consist of a finite sum of terms, each multiplied by a coefficient, whose functional form and number is to be specified by the user. Kernel expansions only need choice of a single kernel function but all eigenfunctions and parametric terms need to be specified as such. While it clearly is impossible to anticipate all possible formulations we note that most (non-kernel) terms can be prepared as separate variables in the GRIDDAT-facility and stored as a separate $x$-variable to enter the specification linearly with a single term. Hence, the linear, single term formulation can in principle address a wide range of formulations. Kernel functions have to be selected as such via the entry FTYPN.

**Likelihood density weight on error.** The choice is between a unitary weight (0) and a census-corrected weight (1). See section 3.7.

**District constraints.** DISTCON activates the imposition of district constraints.

**Factors on regularization, soft margin penalization and window sizes.** These are multiplicative factors on the default values. For regularization, the default values are calculated on the basis of the number of observations in the survey. For window sizes they also depend on the number of explanatory variables.

**Monotonic mapping on dependent variable.** Monotonic mappings may be practical to keep the predictions within a specified range, which may be relevant especially for semiparametric forms, since the non-parametric term does not allow for a priori restrictions on coefficients. The dependent variable is mapped prior to regression, say, by taking a logarithm, and the prediction performs the inverse mapping (for the logarithm by taking the exponent). Mappings can be imposed for all types of regression except kernel smoothing where it is less appropriate.

For the aggregation to district level in Task 5, a lower and an upper bound on $x$, are needed (see section 5 above). Furthermore, scenario simulation can be conducted via a district-specific shifter on $x$.

```plaintext
PARAMETERS
  XLO(R,KX)
  XUP(R,KX)
  XSHIFT(R,KX)

  XLO(R,KX)  =  -9999;
  XUP(R,KX)  =  -9999;
  XSHIFT(R,KX) =  0.;
```

Note: for XLO and XUP = -9999, means no bound; XSHIFT = 0 keeps the independent variables at original level. All three should be specified as the same district level as the other district wise data (here R). Other shifters on the independent variables can be imposed via the GRIDDAT-facility.

**B.6 Selection of the statistics for mapping and tabulation**
Depending on the type of regression conducted, different statistics can be extracted. The user can select from this list. The request will be fulfilled if it is well defined under the requested mode of estimation, and skipped otherwise.

```
SET MAPITEMS 'list of items for mapping and function plotting'

YTOTAL      {1} 'prediction                                   '
YPARAM      {2} 'parametric part of prediction                '
YNONPA      {3} 'nonparametric part of prediction             '
SDEV        {4} 'standard deviation                           '
SURLIK      {5} 'likelihood x wrt survey data                 '
TLIKR       {6} 'likelihood total prediction within b-range   '
PLIKR       {7} 'likelihood param. prediction within b-range  '
TPROBR      {8} 'prob. total prediction within b-range        '
PPROBR      {9} 'prob. param. prediction within b-range       '

Items 1-3 are self-evident, except that jointly they only apply for SV-regression options, as these possess both parametric and a non-parametric part. OLS and mollifier only accept YTOTAL, since they have a parametric and a non-parametric, respectively.

Item 4 is only computed when bagging (with more than one bag) is specified, and not for the mollifier.

Item 5 applies to all regressions and is computed as the likelihood density of kernel smoothing with radial basis, for survey observations only.

Items 6-9 are as defined in section 3.6, where 7 and 9 only apply for SV-regression.

```

Outcomes are stored on gridfiles that can be used in further processing, e.g. as independent variables in conduct recursive regression (instrumentalization), or for gridding. Options 6-9 need specification of the b-range.

```
Aggregate outcomes (in Tasks 3 and 4)

All the extractions of results so far lead to predictions at every observation in the census/grid. If geographical maps are available, aggregations to district average can be invoked in Task 4 through the AV-operation with qualifier DISTRICT to be specified below for calculations over census, as average over census with scaling variable as weight (WEIGHTED), or unity (UNWEIGHT) otherwise. The qualifier TOTALGEN in addition initiates calculation of overall means. The outcomes are stored in GAMS-readable form on the file WKINP\clprsrv.gms.
Furthermore, in Task 3 calculations can be conducted over survey data. These calculations are overall and not by district, because survey data might not carry a district reference. Specifically, overall averages can be taken over survey points, with unitary weights (UNWEIGHT) or weights derived from the census/grid (WEIGHTED). In addition, in Task 3, overall r-squares as realized over the survey can be computed (RSQUARE). Under bagging, it also is possible to compute the average $\beta$ as well as its standard deviation (for $\alpha$-coefficients this is not meaningful). The outcomes are stored in GAMS-readable form on the file \texttt{WKINF\rgprsrv.gms}. The sets below define and activate these operations:

```
SET GENCAL ' general calculations '
   /
   AVERAGE  (1) 'average on predictions ',
   RSQUARE  (2) 'r-square on survey ',
   BETAAV   (3) 'average beta-value under bagging ',
   BETASTD  (4) 'standard deviation beta under bagging ',
   SRV_AVY  (5) 'average over y in survey data ',
   SRV_AVX  (6) 'average over x in survey data ',
   SRV_AVXI (7) 'average over xi at survey points ',
   SRV_AVXIN(8) 'average over net xi at survey points ',
   SRV_STDY (9) 'standard dev y in survey data ',
   SRV_STDX (10) 'standard dev x in survey data ',
   SRV_STDXI(11) 'standard dev xi at survey points ',
   SRV_STDXIN(12) 'standard dev net xi at survey points ',
   SRV_WEIGHT(13) 'census weight at survey points ',
   /;

SET DOGENCAL(GENCAL) ' select general calculations '
   /
   AVERAGE  (1) 'average ',
   RSQUARE  (2) 'r-square on survey ',
   /;

SET GENMOD ' mode of general calculations '
   /
   DISTRICT (1) 'by district ',
   TOTALGEN (2) 'total over districts ',
   UNWEIGHT (3) 'unweighted ',
   WEIGHTED (4) 'weighted by scaling variable ',
   /;

SET DOGENMOD(GENMOD) ' select mode of general calculations '
   /
   DISTRICT (1) 'by district ',
   TOTALGEN (2) 'total over districts ',
   UNWEIGHT (3) 'unweighted ',
   WEIGHTED (4) 'weighted by scaling variable ',
   /;
```

where unweighted calculations also maintain the zero-remains-zero rule, by excluding points with zero scaling weight.

\textbf{B.7 Function plot}
In Task 7, the estimated class wise functions $f_h(x)$ can be plotted. This task needs separate control input, written by REGDAT.GMS on a separate file PLCONTR.TXT, that strictly obeys the specifications laid out in Albersen et al. (2003, section 3), a user guide for plotting a regression function obtained through kernel smoothing but here applies irrespective of the estimation technique used in regression. By editing this file the user can proceed in the way described in the user guide. Here we describe the GAMS-control to generate such a file, with summary explanation only as further documentation can be found in the user guide.

The plotting facility allows for at most one function plot per call of the regression job. There is in a plot in 3D one main function, $f^a$, represented in 3D, and 2 covariates, $f^b$ and $f^c$, represented through coloring of the surface of $f^a$ and coloring of the ground plane. The order in the sequence of dependent variables selected (KFDIST in the example of Appendix A) determines the format of the plot.

$$y^a = f^a(x_1,x_2,z) \text{ surface shape}$$
$$y^b = f^b(x_1,x_2,z) \text{ surface coloring}$$
$$y^c = f^c(x_1,x_2,z) \text{ coloring groundplane}$$

The results are shown for fixed $z$, the independent variables other than $x_1$ and $x_2$. Hence, the ordering of dependent and independent variables provides the key characterization of the graph. We also specify three further sets of options:

```plaintext
SET PLDEF 'Plot definition ' /
  ORDVAR 'Order of variables ' /
  FLAG   'Flag ' /
  PMIN   'Lower bound ' /
  PMAX   'Upper bound ' /

TABLE PLOTY(MAPITEMS,KFDIST) 'selection of variate and covariates '
  MAIZE    PORK
  YTOTAL   1   0
  YPARAM   2   3
  YNONPA   0   0
  SDEV     0   0
  SURLIK   0   0
  TLIKR    0   0
  PLIKR    0   0
  TPROBR   0   0
  FPROBR   0   0
```

The use of the various elements will be explained below. The order of variables is specified by the matrices PLOTY and PLOTX. In PLOTY, we enter 0 if no plotting is required, 1 for $f^a$, 2 for surface co-variate $f^b$ and 3 for plane covariate $f^c$. All three should be specified for well defined statistics.
SDEV is only supported for BAGNR>1, and not for the mollifier. The names PLOTX, PLOTY, PLDEF, ORDVAR are mandatory. The selection is irrespective of the entries in DOMAPITEM.

Similarly, PLOTX has 0-entry if no plotting, 1 for $x_1$, 2 for $x_2$, and 3, 4... for the fixed elements of $z$. The naming PLOTX, PLOTY, PLDEF and ORDVAR is mandatory.

| TABLE PLOTX(PLDEF,KX) "specification for independent variables " |
|---------------------|---------------------|---------------------|
| ORDVAR             | KXX2L | KXX3L | KXX4L |
| FLAG               | 0    | 0    | 0    |
| PMIN               | 0    | 0    | 0    |
| PMAX               | 1    | 1    | 1    |

The order of independent variables is specified in the first row of the PLOTX-matrix. The FLAG-entries (0 or 1) specify, jointly with the scalars PMIN and PMAX, the fixing of the $x_1$, $x_2$ and $z$-variables.

SET PLDEF 'Plotting definition' /
  blank '0 = use PMIN-PMAX percentiles in range [0.,-1.] '
  a    '1 = use average '
  h    '2 = use hard range [PMIN,PMAX] '
  r    '3 = range is [(XMIN+PMIN*DX),(XMAX+PMAX*DX)] '
/;

where XMAX and XMIN are the extremes of the x-observations and DX=(XMAX-XMIN). Hence, under blank-flag we use a percentile range for $x_1$, $x_2$ and a mean over these percentiles to fix the $z$-variables. For $x_1$, $x_2$ under the a-flag, we use the PMIN and PMAX as factors on the mean to determine the range; under h-flag, we use the PMIN and PMAX as hard bounds the range, and under the r-flag we also use the x-range itself. For $z$-variables we only use PMIN.

Next, the set PL2D3D specifies the options for choosing between 2D and 3D plotting

SET PL2D3D 'Possible 3D and 2D plots' /
  3D    ' 3D plot with free covariates '
  3D_P  ' 3D plot with priority for DOVARTYP as covariates '
  mainy ' 2D coloring of main y '
  surface ' 2D coloring of first covariate '
  plane  ' 2D coloring of second covariate '
  all3sep ' 2D main y, surface and plane as separate outputs '
  all3one ' 2D main y, surface and plane in single outputs '
  * additional qualifier (optional)
  rotate ' rotate axis '
/;

Under all3sep, the files will have the name indicated with suffixes _z,_s,_p for main, surface and plane, respectively. For example, 2D of main without rotate will be specified as

SET DO2D3D (PL2D3D) 'Choice of tilts'
/
mainy
/;

Rotation is obtained by:

SET DOPL2D3D (PL2D3D) 'Choice of plot'
/ mainy rotate /;

For a 3D-plot, one uses

SET DOPL2D3D (PL2D3D) 'Choice of tilts'
/ none /

The final stage of the specification is to set three parameters for plotting: the grid size NGRID1, NGRID2 for $x_1, x_2$ that fixes the mesh, say $500 \times 400$ of points at which function evaluation takes place, and PREC that can be 0 or 1, depending on whether to use the previous PLCNTR-file or to create a new one.

SCALARS NGRID1, NGRID2, CONTROLP;

NGRID1 = 500;
NGRID2 = 400;

**B.8 Survey and district data**

Survey and district data can be entered as tables for classes defined by a user-specified index (here we use the same KFDIST as in Appendix A), and a user-specified index set (here KRS):

SET KRS 'observations'
/ KOBS1 KOBS2 KOBS3 KOBS4 /;

TABLE KYV(KRS,KFDIST) "y-observations"

<table>
<thead>
<tr>
<th>KOBS1</th>
<th>MAIZE</th>
<th>PORK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.</td>
<td>101.</td>
</tr>
<tr>
<td>KOBS2</td>
<td>9.</td>
<td>1.</td>
</tr>
<tr>
<td>KOBS3</td>
<td>1.</td>
<td>31.</td>
</tr>
<tr>
<td>KOBS4</td>
<td>12.</td>
<td>21.</td>
</tr>
</tbody>
</table>

TABLE KXR(KRS,KX) "x-observations"
Here survey observations \( KYV \) and \( KXR \) have observation index set \( KRS \). The scaling variable \( KXX1L \) does not appear in this table, since the scaling variable is not a regressor.

Survey and district data are to be entered as tables in the GAMS-job or separately on the file \( WKINP\SRVDAT.TXT \) in the same format as produced by the GAMS-job. The line \( NDATX = -9999; \) should be inserted to indicate separate entry.

Non-available observations are to be entered as \(-9999;\) non-available data for any dependent variable (say, an element of \( KFDIST \)) element will be considered non-available for all elements of the index set i.e. for all elements in that observation. Clearly, the independent variables should be compatible with the grid files specified in B.2. As mentioned in Section 3 survey data will usually have to be expressed in per capita or per hectare terms of the first independent variable, here \( KXX1L \). Yet, the user has to ensure that variables are expressed appropriately, since independent variables may not all be expressed in the same way. For example, if a commodity price enters as independent variable, further division by population or hectares is inappropriate.

The intercept is defined via the matrix \( KXX \) above. The current version has no provision to impose bounds on \( \alpha \) and \( \beta \)-coefficients. However, this and other modifications are readily included in the GAMS program that actually conducts the SV-regression under Task 2 (RGvBag.gms).

Next, to initiate a regression we may assign the numerical values of the district totals. We proceed with the same variables as in Appendix A, assuming that \( CONSQT97 \) is available from an INCLUDE-statement of a file containing the relevant table.

\[
RUCONS(R,KFDIST) = \text{SUM}(IR,CONSQT97(IR,KFDIST,R));
URCONS(R,KFDIST) = \text{SUM}(IU,CONSQT97(IU,KFDIST,R));
\]

Non-available district data are to be entered as \(-9999;\) non-available data for any dependent variable (say, an element of \( KFDIST \)) element will be considered non-available for all elements of the index set i.e. for all elements in that district.

### B.9 Execution

The actual command for regression concludes the operations:

```gams
$OUTREG.gms RUCONS R KFDIST K 14 5 NLEV KORD KRS KX KXX KIX KJX KYV KXXR
```
where the parameters are:

1. matrix of district data
2. geo-index (county/province/region index R,PV,C; 1st subscript)
3. selection from index set
4. index set (with the long labels)
5. number of positions per variable on output
6. number of decimals
7. number of levels (colors/classes) requested for plot
8. order vector of index set
9. RS index set of micro-observations
10. KX index set indep.var.
11. HXV(KFDIST,KX) list of indep. var.
12. IXV(KFDIST,KX) selection of x for parametric terms
13. JXV(KX) selection of x for non-parametric terms
14. YV(RS,JJ) survey data dependent variables
15. XR(RS,KX) survey data independent variables

Dependent variables are processed in one batch, with a common specification for possible nonlinear transformations on. Only the first batch of the GAMS-program will be processed. The mapped out regression function can be used for subsequent instrumentalization in a subsequent run of the sequence (not within one GAMS-program. Only one OUTREG-command can be given in the regression job, and it cannot be combined with gridding operations.
C. Classification estimation and plotting of maps and functions

The Classdat-facility is also a GAMS-controlled regression and mapping software, that combines district, with survey, census, and map information. As classification is intimately connected to regression, the present section significantly duplicates Appendix B. Yet, important differences in techniques and options available in the package justify a separate, self-contained presentation.

The user writes a short GAMS-program that initiates the following tasks:

1. Preparation of data files for classification (Fortran)
2. Estimation (GAMS)
3. Mapping out of estimated function over grid/census and preparation of SAS-files for plotting of maps (Fortran)
4. Aggregation of individual or grid data to district (or higher) level to produce district map
5. Produce district map showing probability of occurrence of classes
6. Running SAS-programs to produce grid maps.
7. Preparing 2D and 3D-graphs of functions

The DOS-script Classdat.bat is used to execute these tasks. A control list in GAMS specifies the sequence of tasks to be conducted.

C.1 Specification of tasks

```gams
SET TASKS    'Possible tasks' /
            TASK1    'prepare matrices for estimation'
            TASK2    'conduct estimation'
            TASK3    'predict function value and choice on survey data'
            TASK4    'predict at every point in census'
            TASK5    'produce district map'
            TASK6    'produce grid map'
            TASK7    'produce 2-3D function plot' /

SET DOTASKS(TASKS)    'Tasks to be executed' /
            TASK1
            TASK2
            TASK3
            /
```

Tasks are activated by defining appropriate (uninterrupted sequences) as subset, for instance:

```gams
SET DOTASKS(TASKS)    'Tasks to be executed' /
            TASK1
            TASK2
            TASK3
            /
```

Task 3 generates a file with the survey data set in GAMS-readable format, extended with the full prediction, the parametric prediction of the function values, as well as the associated binary choice, for further tabulation in GAMS and for possible use as instrumented variables in

---

20 Obviously, producing a gridmap requires the census file to be geo-referenced.
subsequent estimation and similarly for Task 4. Further controls on tasks are specified in section C.6 below.

### C.2 Options for classification

Various techniques can be invoked for classification:

```plaintext
SET CLTECH 'Techniques for classification' /
  EIGCON  (1) 'SV-eigenfunctions'
  KERCON  (2) 'SV-kernel function'
  NEAREST (3) 'Nearest neighbor'
  MOLL    (4) 'Mollifier'
/

SET
  DOCLTECH 'Choose one technique' /
  KERCON   'SV-kercons'
/
```

where the Support vector (SV)-classifications are semiparametric as in section 4, with the $\beta$-coefficients associated to the parametric terms. Specifically,

- **EIGCON**: Primal Quadratic Program for binary SV-classification, for given eigenfunctions
- **KERCON**: Primal Quadratic Program for binary SV-classification, for given kernel functions
- **NEAREST**: Nearest neighbors, number of neighbors specified as NEIGHB in section C.5 below
- **MOLL**: Mollifier or kernel smoothing classifier

As the nearest neighbor and the mollifier are non-parametric, there is no need for parameter estimation, and all operations are conducted at the stage of prediction.

Unlike semiparametric regression, discrete choice problems come in various formats:

```plaintext
SET DCTYP 'Type of discrete choice' /
  SING   'Single classification & quantile selection'
  BINA   'Binary classification'
  MULTC  'Multiple classification, class wise estimation'
  MULTJ  'Multiple classification, joint estimation'
  RANKF  'Ranking with fixed boundaries'
  RANKV  'Ranking with variable boundaries'
/
```

Usually, single classification only serves to select a number of observations from the survey data, and only runs until TASK3. Not all types of discrete choice are supported under every technique. District data are only processed for SV-classification. The table below shows possible combinations:

<table>
<thead>
<tr>
<th>CLTECH:</th>
<th>SV</th>
<th>Nearest</th>
<th>Mollifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>SING</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BINA</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
MULTC             1          1         1
MULTJ             1          0         0
RANKF             1          0         0
RANKV             1          0         0

For single and binary class problems, submission of a data set with multiple classes will lead to separate regressions for every class, and there will be no selection of a winner among these. See also Section C.5 below for the treatment of unclassified outcomes as in eq.(4.1a)-(4.1b).

Next, we specify the scale of aggregation for Task 5:

SET SCALP      'Possible scales for aggregation and plotting'
    / 
    REGP      'regions'
    PROVP     'provinces'
    COUNP     'counties'
    /; 

SET DOSCALP(SCALP)      'Scale to be plotted'
    / 
    COUNP     'counties'
    /;

C.3 Specification of grid and census files

Grid maps or census data are listed in the set REFER as in Appendix A above but since in classification these files are exclusively used to store the independent variables for the district means, we only need one list.

REFER      'Reference variables'
    / 
    F0        ' 1 zero'
    F1        ' 2 one'
    RURP      ' 3 rural'
    URBP      ' 4 urban'
    TOTP      ' 5 total'
    POTY      ' 6 potential yield'
    / 

The file labels (excluding the suffix .grd that will be appended later on) should differ in their first four characters (used for file name) and may be up to 20 characters long.

C.4 Selection of independent variables

Independent variables are listed in the index set KX.

SET KX      'independent variable'
    / 
    KXX1 
    KXX2 
    KXX3 
    KXX4
The matrix \( \mathbf{KXX} \) associates a consolidated list of independent variables for all classes to files in \( \text{REFER} \). The first column entry always has to refer to the scaling variable \( \delta^h \):

\[
\text{TABLE KXX(FUNCTION,KX) "pointers of independent variables to files "}
\]

\[
\begin{array}{ccc}
\text{FUNCTION} & \text{KXX1L} & \text{KXX2L} & \text{KXX3L} \\
3 & 3 & 4 \\
\end{array}
\]

The intercepts of functions should not appear in the first column and are recognized as references to the file \( \text{F1} \) in the set \( \text{REFER} \).

When applicable, the eigenfunctions use independent variables selected from the list \( \text{KX} \) for every dependent variable separately. These variables enter functional forms to be specified in the next section:

\[
\text{TABLE KIX(KFDIST,KX) "select x for eigenfunctions "}
\]

\[
\begin{array}{ccc}
\text{MAIZE} & \text{KXX1L} & \text{KXX2L} & \text{KXX3L} \\
1 & 0 & 1 \\
\text{PORK} & 1 & 2 & 1 \\
\end{array}
\]

Here a zero entry means non-selection, and a positive entry establishes the number of terms in a function expansion (e.g. a polynom) in which the variable should enter. Similarly, for the parametric terms the independent variables are selected as:

\[
\text{TABLE KJX(DEFINE,KX) "select x for parametric terms "}
\]

\[
\begin{array}{ccc}
\text{MAIZE} & \text{KXX1L} & \text{KXX2L} & \text{KXX3L} \\
1 & 1 & 0 \\
\text{PORK} & 1 & 0 & 1 \\
\end{array}
\]

Kernel functions will contain as arguments all independent variables except the scaling variable and the intercept.

**C.5 Control parameters for classification: district data, bagging, window size, functional form**

The control parameters for classification are defined in the set \( \text{CLASSCONT} \) and the table \( \text{CONTVAL} \):

\[
\text{SET CLASSCONT ' control parameters for regression '}
\]

\[
\text{BAGNR ' number of bags}
\]
Bagging. To assess the stability of estimates, bagging can be conducted whereby the survey data set is subdivided into an equal number \((BAGNR)\) of subsets for which estimation is conducted separately, yielding subset-specific predictions. The bags can be built (a) by straight subdivision of the sample into separate parts \((BAGREP=0)\); or (b) by sampling with replacement from the full data set \((BAGREP=1)\). The prediction used is the mean over these predictions, and when bagging is active a standard deviation can be computed as well.

Function type. The parametric part and the nonparametric part consist of a finite sum of terms, each multiplied by a coefficient, whose functional form and number is to be specified by the user. Kernel expansions only need choice of a single kernel function but all eigenfunctions and parametric terms need to be specified as such. While it clearly is impossible to anticipate all possible formulations we note that most (non-kernel) terms can be prepared as separate variables in the GRIDAT-facility and stored as a separate \(x\)-variable to enter the specification linearly with a single term. Hence, the linear, single term formulation can in principle address a wide range of formulations. Kernel functions have to be selected as such via the entry \(FTYPN\).

Likelihood density weight on error. The choice is between a unitary weight \((0)\) and a census-corrected weight \((1)\). See section 3.7.

District constraints. \(DISTCON\) activates the imposition of district constraints; \(DISTGAM\) is the error tolerance \(\gamma\) on the district constraint as in (4.17).
Factors on regularization, soft margin penalization and window sizes. These are multiplicative factors on the default values. For regularization, the default values are calculated on the basis of the number of observations in the survey. For window sizes they also depend on the number of explanatory variables (see section 3).

Unclassified. In multiple classification can follow (4.1a) or (4.1b) depending on whether we require the function value to be positive. Clearly, multiple classification with joint estimation (MULTJ) will always generate a positive value, unless the problem is degenerated with zero function value for all classes.

The ranking problem with fixed bounds (RANKF, see section 4.6) needs bounds on $y$. The first and the last bound enclose the range of the $y$-function. Hence, the number of bounds exceeds the number of classes by one. The classes are denoted here by KFDIST:

```plaintext
SET BNDY 'bounds for discrete choice' / 
  BNDY01     'Bound 1' 
  BNDY02     'Bound 2' 
  BNDY03     'Bound 3' 
/;

TABLE BRACKETY(BNDY,INIT) 'Upper bounds on y'

     VALUE
   BNDY01       0. 
   BNDY02       2. 
   BNDY03      100. 
;
```

For the aggregation to district level in Task 5, a lower and an upper bound on independent variables $x$, are needed (see Section 5 above). Furthermore, scenario simulation can be conducted via a district-specific shifter on $x$.

```plaintext
PARAMETERS
  XLO(R,KX) 
  XUP(R,KX)
  XSHIFT(R,KX)
;

  XLO(R,KX) = -9999;
  XUP(R,KX) = -9999;
  XSHIFT(R,KX) = 0.;
```

Note: for XLO and XUP = -9999. means no bound; XSHIFT = 0 keeps the independent variables at original level. All three should be specified as the same district level as the other district wise data (here $R$). Other shifters on the independent variables can be imposed via the GRIDAT-facility.

C.6 Selection of the statistics for mapping and tabulation
Depending on the type of classification conducted, different statistics can be extracted. The user can select from this list. The request will be fulfilled if it is well defined under the requested mode of estimation, and skipped otherwise.

```
SET MAPITEMS 'list of items for mapping and function plotting'

YTOTAL    {1} 'prediction'
YPARAM    {2} 'parametric part of prediction'
YNONPA    {3} 'nonparametric part of prediction'
SDEV      {4} 'standard deviation'
SURLIK    {5} 'likelihood x wrt survey data'
TLIKR     {6} 'likelihood total prediction within b-range'
PLIKR     {7} 'likelihood param. prediction within b-range'
TPROBR    {8} 'prob. total prediction within b-range'
PPROBR    {9} 'prob. param. prediction within b-range'

Items 1-3 are self-evident, except that jointly they only apply for SV-classification options, as these possess both parametric and a non-parametric part. OLS and mollifier only accept YTOTAL, since they have a parametric and a non-parametric, respectively.

Item 4 is only computed when bagging (with more than one bag) is specified, and not for the mollifier.

Item 5 applies to all classifications and is computed as the likelihood density of kernel smoothing with radial basis, for survey observations only.

Items 6-9 are as defined in section 3.6, where 7 and 9 only apply for SV-classification.

For ranking, applying functions (4.20) makes it possible to present results by class, despite the fact that there only is a single function estimated for all classes.

```
SET DOMAPITEMS(MAPITEMS) 'selection of items to be mapped'

YTOTAL    {1} 'prediction'
YPARAM    {2} 'parametric part of prediction'
YNONPA    {3} 'nonparametric part of prediction'
SURLIK    {5} 'likelihood x wrt survey data'
TLIKR     {6} 'likelihood total prediction within b-range'
PLIKR     {7} 'likelihood param. prediction within b-range'
TPROBR    {8} 'prob. total prediction within b-range'
PPROBR    {9} 'prob. param. prediction within b-range'

Outcomes are stored on grid files that can be used in further processing, e.g. as independent variables in conduct recursive estimation (instrumentalization), or for gridding. Options 6-9 need specification of the b-range.

The winning class and the runner up can also be identified and mapped out, via the following index sets:

```
SET MAPCLASS 'possible classifications for mapping and plotting'

WINNERT    {1} 'class h of selected total function'
```
Aggregate outcomes (in Tasks 3 and 4)

All the extractions of results so far lead to predictions at every observation in the census/grid. If geographical maps are available, aggregations to district average can be invoked in Task 4 through the AV-operation with qualifier DISTRICT to be specified below for calculations over census, as average over census with scaling variable as weight (WEIGHTED), or unity (UNWEIGHT) otherwise. The qualifier TOTALGEN in addition initiates calculation of overall means. The outcomes are stored in GAMS-readable form on the file WKINP\clprsrv.gms.

Furthermore, in Task 3 calculations can be conducted over survey data. These calculations are overall and not by district, because survey data might not carry a district reference. Specifically, overall averages can be taken over survey points, with unitary weights (UNWEIGHT) or weights derived from the census/grid (WEIGHTED). In addition, in Task 3, overall hit-ratios as realized over the survey can be computed (HITRATIO). Under bagging, it also is possible to compute the average $\beta$ as well as its standard deviation (for $\alpha$-coefficients this is not meaningful). The outcomes are stored in GAMS-readable form on the file WKINP\clprsrv.gms. The sets below define and activate these operations:

```
SET GENCAL ' general calculations ' /
   AVERAGE    {1} 'average on predictions    '  
   HITRATIO   {2} 'hit ratio on survey      '  
   BETAAV     {3} 'average beta-value under bagging' 
   BETASTD    {4} 'standard deviation beta under bagging' 
   SRV_AVY    {5} 'fraction of yes in survey data'  
   SRV_AVX    {6} 'average over x in survey data'   
   SRV_STDX   {7} 'standard deviation x in survey data'  
   SRV_WEIGHT {8} 'census weight at survey points'  
/
SET DOGENCAL(GENCAL) ' select general calculations ' /
   AVERAGE    {1} 'average on predictions    '  
   HITRATIO   {2} 'hit ratio on survey      '  
/
SET GENMOD   ' mode of general calculations ' /
   DISTRICT   {1} 'by district             '  
   TOTALGEN   {2} 'total over districts     '  
   UNWEIGHT   {3} 'unweighted              '  
   WEIGHTED   {4} 'weighted by scaling variable' 
/;```
where unweighted calculations also maintain the zero-remains-zero rule, by excluding points with zero scaling weight.

### C.7 Function plot

Task 7 plots the estimated class wise functions \( f_i(x) \) as well as the statistics derived from these. This task needs separate control input, written by CLASSDAT.GMS on the file PLCONTR.TXT. This file strictly obeys the specifications laid out in Albersen et al. (2003, section 3), a user guide for plotting a regression function obtained through kernel smoothing but here applies irrespective of the estimation technique used in regression. By editing this file the user can proceed in the way described in the user guide. Here we describe the GAMS-control to generate such a file, with summary explanation only as further documentation can be found in the user guide.

The plotting facility allows for at most one function plot per call of the classification job. There is in a 3D-plot one main function, \( f^a \), represented in 3D, and 2 covariates, \( f^b \) and \( f^c \), represented through coloring of the surface of \( f^a \) and coloring of the ground plane. The order in the sequence of dependent variables selected (index set \( KFDIST \) in the example above) determines the format of the plot.

\[
\begin{align*}
y^a & = f^a(x_1,x_2,z) \quad \text{surface shape} \\
y^b & = f^b(x_1,x_2,z) \quad \text{surface coloring} \\
y^c & = f^c(x_1,x_2,z) \quad \text{coloring ground plane}
\end{align*}
\]

The results are shown for fixed \( z \), the independent variables other than \( x_1 \) and \( x_2 \). Hence, the ordering of dependent and independent variables provides the key characterization of the graph. We also specify three further sets of options:

```plaintext
SET PLDEF 'Plot definition' / 
  ORDVAR 'Order of variables' 
  FLAG 'Flag' 
  PMIN 'Lower bound' 
  PMAX 'Upper bound' 
/;
```

The use of the various elements will be explained below. The plots should be specified for at most three statistics to be selected from the tables PLOTY for class-specific statistics, PLOTX for statistics on the winning class, and PLOTX for the ordering of independent variables. In PLOTY and
PLOTC, we enter 0 if no plotting is required, 1 for \( f^a \), 2 for surface co-variate \( f^b \) and 3 for plane co-variate \( f^c \).

TABLE PLOTY(MAPITEMS,KFDIST) 'selection from class wise results'

<table>
<thead>
<tr>
<th></th>
<th>MAIZE</th>
<th>FORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>YTOTAL</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>YPARAM</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>YNONPA</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SDEV</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SURLIK</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TLIK</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PLIK</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TPROBR</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PPROBR</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SDEV is only supported for BAGNR>1, and not for the mollifier. The names PLOTC, PLOTX, PLOTY, PLDEF, ORDVAR are mandatory.

TABLE PLOTC(MAPCLASS,INIT) 'selection from overall results'

<table>
<thead>
<tr>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WINNERT</td>
</tr>
<tr>
<td>WINNERP</td>
</tr>
<tr>
<td>RUNUPT</td>
</tr>
<tr>
<td>RUNUPP</td>
</tr>
<tr>
<td>YWINT</td>
</tr>
<tr>
<td>YWINP</td>
</tr>
<tr>
<td>YMARGT</td>
</tr>
<tr>
<td>YMARGP</td>
</tr>
</tbody>
</table>

Combined, PLOTY and PLOTC can have at most three non-zero entries. The selection is irrespective of the entries in DOMAPITEM and DOMAPCLASS.

Similarly, PLOTX has 0-entry if no plotting, 1 for \( x_1 \), 2 for \( x_2 \), and 3, 4... for the fixed elements of \( z \). The naming PLOTX, PLOTY, PLDEF and ORDVAR is mandatory.

TABLE PLOTX(PLDEF,KX) "specification for independent variables"

<table>
<thead>
<tr>
<th>ORDVAR</th>
<th>KXX2L</th>
<th>KXX3L</th>
<th>KXX4L</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAG</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FMIN</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>FMAX</td>
<td>1.</td>
<td>1.</td>
<td>1.</td>
</tr>
</tbody>
</table>

The order of independent variables is specified in the first row of the PLOTX-matrix. The FLAG-entries (0 or 1) specify, jointly with the scalars FMIN and FMAX, the fixing of the \( x_1, x_2 \) and \( z \)-variables.

SET PLDEF 'Plotting definition'
(/ blank  '0 = use PMIN-PMAX percentiles in range [0.,-1.] '
   a     '1 = use average '
   h     '2 = use hard range [PMIN,PMAX] '
   r     '3 = range is [(XMIN+PMIN*DX),(XMAX+PMAX*DX)] '
/);

where XMAX and XMIN are the extremes of the x-observations and DX=(XMAX-XMIN). Hence, under blank-flag we use a percentile range for \( x_1, x_2 \), and a mean over these percentiles to fix the \( z \)-variables. For \( x_1, x_2 \) under the a-flag, we use the PMIN and PMAX as factors on the mean to determine the range; under h-flag, we use the PMIN and PMAX as hard bounds the range, and under the r-flag we also use the x-range itself. For \( z \)-variables we only use PMIN.

Next, the set PL2D3D specifies the options for choosing between 2D and 3D plotting

```plaintext
SET PL2D3D 'Possible 3D and 2D plots'
/ 3D      ' 3D plot with free covariates '
3D_P     ' 3D plot with priority for DOVARTYP as covariates '
mainy    ' 2D coloring of main y '
surface  ' 2D coloring of first covariate '
plane    ' 2D coloring of second covariate '
all3sep  ' 2D main y, surface and plane as separate outputs '
all3one  ' 2D main y, surface and plane in single outputs '
* additional qualifier (optional)
   rotate ' rotate axis '
/;
```

Under all3sep, the files will have the name indicated with suffixes _z_,_s_,_p for main, surface and plane, respectively. For example, 2D of main without rotate will be specified as

```plaintext
SET DO2D3D (PL2D3D) 'Choice of tilts '
/  mainy
/;
```

Rotation is obtained by:

```plaintext
SET DOPL2D3D (PL2D3D) 'Choice of plot '
/  mainy
   rotate
/;
```

For a 3D-plot, one uses

```plaintext
SET DOPL2D3D (PL2D3D) 'Choice of tilts '
/  none
/;
```

The final stage of the specification is to set three parameters for plotting: the gridsize NGRID1, NGRID2 for \( x_1, x_2 \) that fixes the mesh, say \( 500 \times 400 \) of points at which function evaluation
takes place, and \texttt{PREC} that can be 0 or 1, depending on whether to use the previous PLCONTR-file or to create a new one.

\begin{verbatim}
SCALARS NGRID1, NGRID2, CONTROLP;
    NGRID1 = 500;
    NGRID2 = 400;
\end{verbatim}

\section*{C.8 Survey and district data}

Survey and district data can be entered as tables for classes defined by a user-specified index (see index set \texttt{KFDIST} in Appendix A), and a user-specified index set (here index set \texttt{KRS}):

\begin{verbatim}
SET KRS 'observations'
/ KOBS1 KOBS2 KOBS3 KOBS4 /;
TABLE KYV(KRS,KFDIST) "y-observations"
   MAIZE   PORK
     KOBS1   1       -1
     KOBS2  -1       1
     KOBS3   1       -1
     KOBS4  -1    -9999
;

TABLE KXR(KRS,KX) "x-observations"
   KXX2L   KXX3L
     KOBS1  1.       1.
     KOBS2  1.       1.
     KOBS3  1.       1.
     KOBS4  1.       1.
;
\end{verbatim}

For single and binary classification defining of two or more classes will initiate separate regressions for different data sets, and columns are independent. For other classifications only one positive value can occur in every row. The observations on independent variables can be entered as follows:

\begin{verbatim}
TABLE KXR(KRS,KX) "x-observations"
   KXX2L   KXX3L
     KOBS1  1.       1.
     KOBS2  1.       1.
     KOBS3  1.       1.
     KOBS4  1.       1.
;
\end{verbatim}

Here survey observations \texttt{KYV} and \texttt{KXR} have observation index set \texttt{KRS}. The scaling variable \texttt{KXX1L} does not appear in this table, since the scaling variable is not a regressor.

Survey and district data are to be entered either as tables in the GAMS-job or separately on the file \texttt{WKINP\$RVDAT.TXT} in the same format as produced by the GAMS-job. The line \texttt{NDATX = -9999;} should be inserted to indicate separate entry.
Non-available observations are to be entered as −9999; non-available data for any dependent variable (say, an element of \( KFDIST \)) element will be considered non-available for all elements of the index set i.e. for all elements in that observation. Clearly, the independent variables should be compatible with the grid files specified in B.2. As mentioned in section 3 survey data will usually have to be expressed in per capita or per hectare terms of the first independent variable, here \( KXX1L \). Yet, the user has to ensure that variables are expressed appropriately, since independent variables may not all be expressed in the same way. For example, if a commodity price enters as independent variable, further division by population or hectares is inappropriate.

The intercept is defined via the matrix \( XXX \) above. The current version has no provision to impose bounds on coefficients. However, this and other modifications are readily included in the GAMS program that actually conducts the SV-classification under Task 2 (CLvBag.gms).

The format of data input for the dependent variables is the same across all options and identical to the format of Appendix B of real-valued dependent variables, except that under single classification all data should have value one, under all other options the binary values \{−1,1\} of (4.2) have to be entered. In fact, the package allows for real-valued input: positive entries will be set to 1, −9999 to undefined, all others to −1. Here for ease of comparison with Appendix B, we use the same categories MAIZE and PORK as classes: MAIZE is class 1, PORK class 2.

Next, to initiate a classification we may assign the numerical values of the district means. We proceed with the same variables as in Appendix A, assuming that \( CONSQNT97 \) is available from an INCLUDE-statement of a file containing the relevant table.

PARAMETERS
    RUCONS(R,KFDIST) "Voting shares rural"
    URCONS(R,KFDIST) "Voting shares urban"
;
    RUCONS(R,KFDIST) = SUM(IR,CONSQNT97(IR,KFDIST,R));
    URCONS(R,KFDIST) = SUM(IU,CONSQNT97(IU,KFDIST,R));

Non-available district data are to be entered as −9999.; non-available data for any dependent variable (say, an element of \( KFDIST \)) element will be considered non-available for all elements of the index set i.e. for all elements in that district.

C.9 Execution

The actual command for regression concludes the operations:

\$
...OUTCLASS.gms RUCONS R KFDIST K 14 5 NLEV KORD KRS XX KXX KIX KJX KYV KXXR
\$

where the parameters are:

1. matrix of district data
2. geo-index (county/province/region index R,PV,C; 1st subscript)
3. selection from index set
4. index set (with the long labels)
5. number of positions per variable on output
6. number of decimals
7. number of levels (colors/classes) requested for plot
Dependent variables are processed in one batch. Only the first batch of the GAMS-program will be processed. The mapped out function can be used for subsequent instrumentalization in a subsequent run of the sequence (not within one GAMS-program. Only one OUTCLASS-command can be given in the classification job, and it cannot be combined with gridding operations.
D. Directory structure and data input-output

The package expects all applications to obey the same directory structure, whose main directory (with a blank first line and e.g. C:\CHINAPART\ on the second line) is specified in the files WKINP\FADDR.TXT in the main GAMS-program.

Main directory
- DAT Data files (.grd: grid/census files; .gms: district files)
- SRC Main programs (GAMS sources and Fortran executables)
- LIBRARY GAMS subroutines
- MAKEMAP .gif files produced through SAS with maps and plots
- SASDAT data files with maps for use by SAS-macros
- SASJOBS subdirectory to keep user written SAS-jobs
- MACROS SAS-macros for plotting
- WKINP Input files created during execution for reference by user
- WKOUT Output files
- WKRUN Temporary files

Input files

Three types of application-specific input files can be distinguished.

(a) GIS segment-files that contain the line segments of the polygons of the map and administrative boundaries (county/district, province, region/nation) for the area under consideration. The polygon files are extracted from freely available GIS-databases. They are the only data files in the package that are not to be supplied by the user. They are located in MAKEMAP\SASDAT, and readable by the SAS-macros (subroutines) for plotting at grid, county, province or regional level MAKEMAP\SASJOBS\MACROS. Users may, jointly with the software, request SOW-VU for a tailor made set of such GIS-files that can subsequently be used for any application referring to this country-(group), see Overbosch (forthcoming).

(b) A single ASCII-file (DAT\locat.grd) also tailor made for this country-(group) indicating to which county, province and region every grid cell (or census-household) belongs, and whether it is on mainland or sea. The grid cell file is supplied jointly with the software and starts with the header:

* maximal latitude
  [integer number]
* maximal longitude
  [integer number]
* number of records
  [integer]

directly followed by records with the following integers (hence without characters):

iden lat lon county province region mainlandcode

The mainland code is 0 for searoutes and other parts of the sea to be represented and 1 for mainland cells. The user does not have to provide entries for cells at which no calculations are to
take place (e.g. unpopulated cells without transport links). The records are to be sorted by increasing order of the first entry (iden). The lat lon entries should be within the range \([0, \text{maximal latitude}]\) and \([0, \text{maximal longitude}]\) and the number of records should be as specified. For geographic maps it refers to the number of grid cells, for census to the number of households in the data set.

(c) files with a single gridded variable: these are the REFERINI-files as defined in the GAMS-application as independent variables and variables to be gridded. They have .grd extension are located in DAT and only contain one real number per record. Grid cells (households) should be ranked in the order specified in the locat.grd file. They can be created by overlaying the lat-lon of an original GIS-data set with this file, only keeping the entries appearing on locat.grd, while setting the locat.grd-entries not appearing in the data equal to zero (hence the zero-remains-zero restriction).

(d) GAMS-files with district data (at county province and region level). These data are supplied as matrices on GAMS-files (.gms) from a location specified by the include statement in the GAMS-program. In every matrix the geographical index appears in the rows, and with \(\text{CN}[\text{integer}], \text{PV}[\text{integer}]\) and \(\text{R}[\text{integer}]\) as identifiers (elements of index sets).

(e) Intermediate input files are output from a previous step and stored in WKINP and WKRUN, depending on whether the user will expectedly consult this file or not.

Output files

(a) The grid/census-files created by the GRIDDAT, REGDAT and CLASSDAT facilities obey the same structure as the input files under (c), and are, therefore, useable across applications. The files in REFERNAM for which storage is requested under the mapping of GRIDDAT are given the four-character names specified and stored in the subdirectory DAT. In addition, if plotting of maps is requested (CONTROLP = 1 or 2), grid files in Excel (extension .csv) with acronyms built up from the index-set names and entries, and the scenario names are stored in WKRUN, and if in addition storage as grid files is requested the same data are stored grid files under the same name in subdirectory DAT and with .grd extension.

(b) Log-files with debugging information can be found in WKOUT for output from the Fortran-programs and in WKRUN for output from SAS.

(c) Maps are stored in MAKEMAP\PICT with a .gif extension, and can be browsed through by XNVIEW-software, or any other picture-viewer. Zooming in can be used, in particular for grid maps, as their resolution is usually far larger than the pixels shown on a screen, to focus on particular zones.

Specifically, the following files are (automatically) written by the program, for GRIDDAT.GMS:

WKOUT\GRIDDAT.LOG:  log
WKINP\VARINP.TXT:   input for further processing

for REGDAT.GMS:

WKOUT\REGDAT.LOG:  log
WKINP\PLCONTR.TXT:  control file for plotting
WKINP\RGPRCEN.GMS:  GAMS-readable output from function evaluations after aggregation to district level over census for further processing and tabulation
WKINP\RGPRSRV.GMS:  GAMS-readable output from function evaluations at data points of survey for further processing and tabulation
WKINP\RGINP.TXT:    input from REGDAT.GMS for further processing
WKINP\RGPAR.TXT:    estimated coefficients
WKINP\SRVDAT.TXT:   survey data
WKRUN\RGvBAG.LST:   GAMS-log from estimation

and for CLASSDAT.GMS:

WKOUT\CLASSDAT.LOG: log
WKINP\CLINF.TXT:    input from CLASSDAT.GMS for further processing
WKINP\CLPAR.TXT:    estimated coefficients
WKINP\CLPRCEN.GMS:  GAMS-readable output from function evaluations after aggregation to district level over census for further processing and tabulation
WKINP\CLPRSRV.GMS:  GAMS-readable output from function evaluations at data points of survey for further processing and tabulation
WKINP\PLCONTR.TXT: control file for plotting
WKINP\SRVDAT.TXT:   survey data
WKRUN\CLvBAG.LST:   GAMS-log from estimation
References


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 – 598 9321
Telefax (31) 20 – 598 9325
Email pm@sow.vu.nl
www http://www.sow.vu.nl/