# A Practical Guide to Geostatistical Mapping of Environmental Variables

Tomislav Hengl
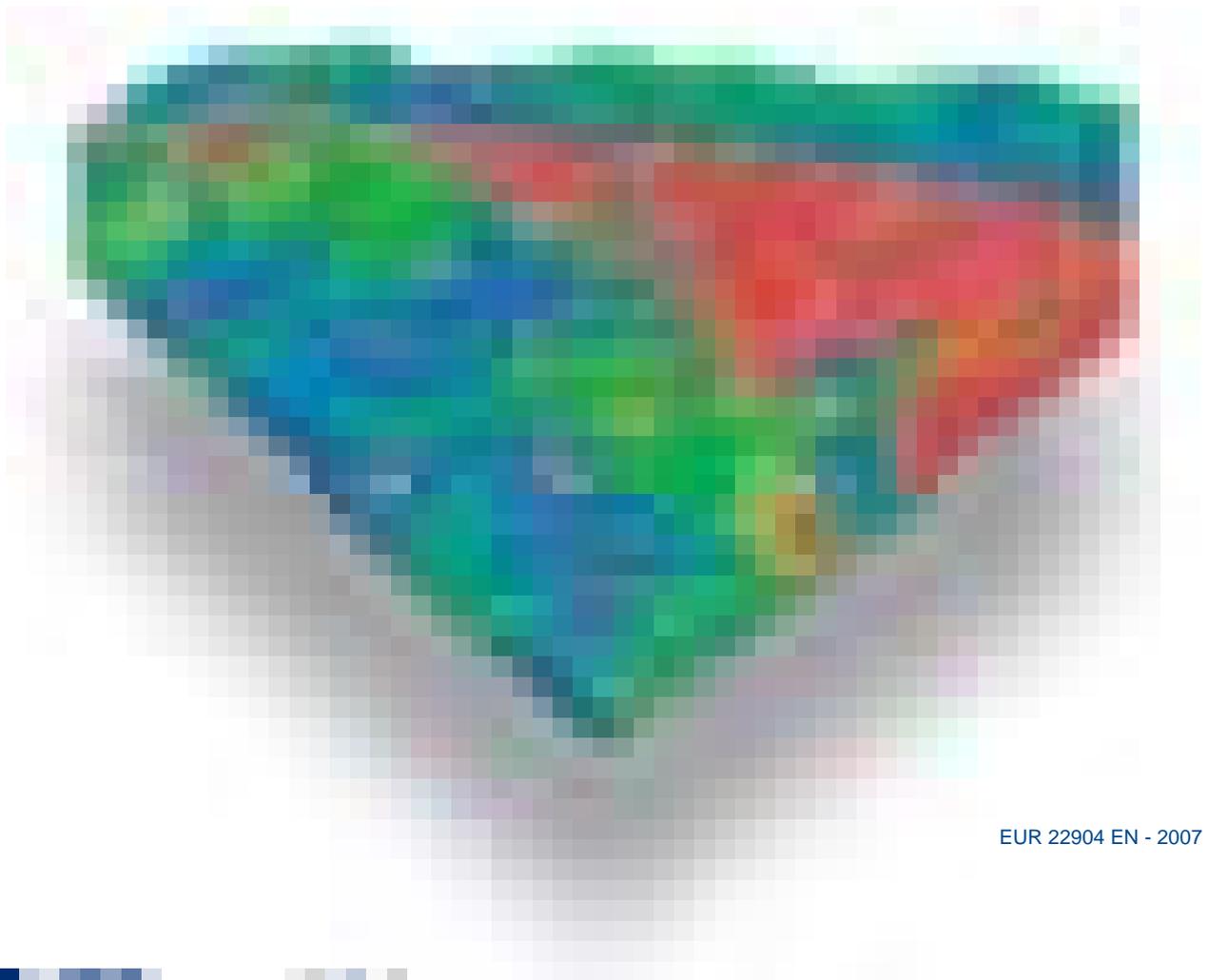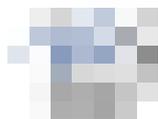
The mission of the Institute for Environment and Sustainability is to provide scientific-technical support to the European Union's Policies for the protection and sustainable development of the European and global environment.

Printed in Italy

Abstract

Geostatistical mapping can be defined as analytical production of maps by using field observations, auxiliary information and a computer program that calculates values at locations of interest. Today, increasingly the heart of a mapping project is, in fact, the computer program that implements some (geo)statistical algorithm to a given point data set. Purpose of this guide is to assist you in producing quality maps by using fully-operational tools, without a need for serious additional investments. It will first introduce you the to the basic principles of geostatistical mapping and regression-kriging, as the key prediction technique, then it will guide you through four software packages: ILWIS GIS, R+gstat, SAGA GIS and Google Earth, which will be used to prepare the data, run analysis and make final layouts. These materials have been used for the five-days advanced training course "Hands-on-geostatistics: merging GIS and spatial statistics", that is regularly organized by the author and collaborators. Visit the course website to obtain a copy of the datasets used in this exercise.

The mission of the JRC is to provide customer-driven scientific and technical support for the conception, development, implementation and monitoring of EU policies. As a service of the European Commission, the JRC functions as a reference centre of science and technology for the Union. Close to the policy-making process, it serves the common interest of the Member States, while being independent of special interests, whether private or national.
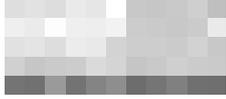
A Practical Guide to Geostatistical Mapping
of Environmental Variables

by T. Hengl

September 2007

Contents

iv

Foreword

An impression I had over the years, while working for various digital soil mapping projects, is that there is a serious gap between what is known by few (researchers) and what is actually implemented in practice (users). On one hand, we have sophisticated the ways to produce more and more detailed/informative maps, on the other hand the users rely on traditional mapping systems. This seems to be a gap between the users and tools rather than a gap in the theory. In the last few years, things have started improving rapidly. First, tools that allow merging of GIS and (geo)statistical operations have been made operational and available to many. Second, there is an increase of free remote sensing (e.g. MODIS) and relief data (e.g. SRTM DEM), which are available at global scale at resolution of 250 m or finer (see further Table 3.2). And third, many processing steps can now be automated, which makes it possible to run computations using extensive and complex databases. Now many environmental agencies have to catch up with this rapid advances of both technology and software. Only within JRC Ispra there are several mapping and monitoring projects — BIOSOIL, LUCAS, Geochemical Atlas of Europe, INTAMAP, Danube Basin — that now completely rely on the availability of such semi-automated mapping tools.

The main purpose of this guide is to assist you in using geostatistical tools with your own data. You are now invited to produce quality maps by using fully-operational tools implemented in an open-source software. I advocate the band of four: ILWIS, R+gstat, SAGA GIS and Google Earth. There are probably several alternatives on the market, however, the arguments are clear: (1) all four are available as open-source or as freeware; (2) all alow scripting (data processing automation) and extension of existing functionality, and (3) all support data exchange through GDAL and similar engines. I assume that your experience with using open source packages was probably very frustrating, because many provide only command-line interface and the commands follow some particular philosophy for which there is a limited support. However, my experience with for example R is that, after one learns the basic steps and ways to get support and more explanation of algorithms, it is a steep learning curve. My intention with this handbook was similar – I wanted to assist you in obtaining the software and making the first steps, warn what might be the bottlenecks and what you should avoid doing, and provide the most crucial tricks'n'tips on how to build scripts and organize the data processing. Ideally, you should be able to generate maps from your point datasets and interpret the results, just by following this guide.

The guide consists of four chapters. The first chapter is an introductory chapter to the practice of geostatistical mapping and gives an overview of the spatial prediction techniques. The second chapter zooms into regression-kriging and its characteristics, advantages and limitations. The third chapter is completely dedicated to installation and doing first steps in the software, and the last, fourth, chapter gives a step-by-step guide through analysis and generation of final layouts by using a digital soil mapping case study. After reading the first chapter, you should understand what the geostatistical mapping is; after reading the second chapter, you should know how to select the right spatial prediction technique for your application; after reading the third chapter, you should be able to install all packages used in the handbook and be aware of their capabilities; and after reading the fourth chapter, you should know how to run geostatistical mapping, prepare final layouts and interpret the results of analysis for your own case study.

This guide evolved as a lecturing material that has been used for a 5-day training course called "*Hands-on-geostatistics: Merging GIS and Spatial Statistics*". The objective of this course was to provide theoretical backgrounds and practical training on the use of hybrid geostatistical/GIS tools for various applications ranging from spatial prediction to sampling and error propagation. In addition, the *leitmotive* of the course was to provide practical training in command-based software packages such as R. We aimed at Master and PhD level students and post-doctoral researchers in various fields of environmental and geo-sciences interested in spatial prediction and analysis of environmental variables. We have run this course already twice: at the *Facolta di Agraria* in Naples (29.01-03.02.2007), and at JRC Ispra (03.06-07.06.2007). At both occasions, the interest exceeded our expectations. In fact, many course participants complained that their previous geostatistics courses focused too much on plain geostatistics (pure theoretical training) or were based on commercial packages (e.g. Isatis, ArcGIS). In our case, about 40% of the course has been dedicated to work in open-source software and practical aspects of data analysis: it included training on how to build and edit scripts in R and ILWIS, how to use commands in gstat and sp packages, how to export GIS layers to Google Earth and generate final layouts etc. This guide follows more or less the same structure, except it is probably more extensive and one would not be able to teach all these topics within five days.

Many participants of the course repeatedly asked me the same question: "*Can I also use these tools with my own data and are they really for free?*". The answer is definitively: YES! However, I can not guarantee that you can generate quality maps by using low quality field data (please read the disclaimer on p. ix). In other words, nobody can guarantee that your datasets can be *saved* with these tools, so make sure you provide quality inputs. There are certainly limits to what you can do with regression-kriging. For example, you will soon discover that larger datasets are more difficult to process and can lead to computational difficulties. Running spatial prediction of $\gg 10^3$ points using grids of $\gg$1M pixels might last several hours on a standard PC. The computation time will increase exponentially for higher number of input points and finer grid resolutions. Solving such computational cumbersome will be a quest, both for environmental and computer scientists.

Another important motive to produce this handbook was to diminish frustrations a typical beginner has with geostatistical theory. Many users of geostatistics are confused with the amount of methods and with interpreting the results of some computation in a statistical software. I have done my best to try to diminish the terminological confusion (e.g. confusion between universal kriging using coordinates and predictors; confusion

between running local and localized predictions) and warn the users which techniques are valid for use and for which situations. With this handbook, you can now zoom into a certain technique, into the data processing steps that are more interesting for your case studies, and select the optimal methodology that fits your objectives. The rest, we can discuss via the mailing lists.

The author of this user's guide would like to thank people that have contributed to this publication. The first on the long list is definitely Edzer Pebesma from the University of Utrecht, the creator of gstat and one of the most open-minded people that I have met so far. We can all, in fact, thank Edzer for kindly providing the source code and his professional opinions (the gstat mailing list) over the last decade. This document would certainly not exist without his generosity and dedication to the field. The second on the list is David G. Rossiter who assisted me in organizing the course at JRC. David has been contributing to the knowledge of geostatistics through his course on geostatistics that he has been regularly organizing over the years at ITC. He also kindly provided many handbooks and R codes that you can at any time access from his website. The next on the list is my JRC colleague Gregoire Dubois who critically read this document and provided suggestions and useful references. I also feel obliged here to thank Fabio Terrible from the University of Naples for inviting me to organize this course in his town and for hosting us in Naples. Likewise, I need to thank Pernille Brandt, the head manager of the LMU Human Resources, and her assistants for supporting our course at JRC. Many thanks also to participants of the *Hands-on geostatistics* course for their interesting questions and comments that helped shaped this handbook.

The author would also like to thank Gehrt Ernst from the State Authority for Mining, Energy and Geology, Hannover, Germany for providing the Ebergötzen dataset[1] with a full description of its content and lineage. I was truly surprise to discover the amount of geostatistical ideas Erst had already back in 1990s (way before I even heard about geostatistics). I am now slowly refreshing my German by studying the documents Ernst forwarded.

From this point on, I will use a matrix notation to describe computational steps, which is often not easy to follow by a non-mathematician. For an introduction to matrix algebra, read the general introductions in classical statistical books such as Neter et al. (1996, §5). A detailed introduction to matrix algebra used in geostatistics can be also found in Wackernagel (2003). Finally, I should note that this handbook definitively does not offer a complete coverage of the field of geostatistics and readers are advised to extend their knowledge by obtaining the literature listed at the end of each chapter or as referred to in the text. The terminology used in this handbook and many statements are purely subjective and can be a subject of discussion.

Every effort has been made to trace copyright holders of the materials used in this book. The European Commission apologizes for any unintentional omissions and would be pleased to add an acknowledgment in future editions.

Tomislav Hengl

Ispra (VA), September 2007

---

[1] The Ebergötzen datasets, scripts and codes used in this handbook can be obtained from the course website http://geostat.pedometrics.org.

Disclaimer

All software used in this guide is free software and comes with ABSOLUTELY NO WARRANTY. The information presented herein is for informative purposes only and not to receive any commercial benefits. Under no circumstances shall the author of this Guide be liable for any loss, damage, liability or expense incurred or suffered which is claimed to resulted from use of this Guide, including without limitation, any fault, error, omission, interruption or delay with respect thereto (reliance at User's own risk).

The readers are advised to use the digital PDF version of this document, because many URL links are embedded and will not be visible from the paper version. You are welcome to redistribute the programm codes and the complete document provided under certain conditions. For more information, read the GNU general public licence.

The main idea of this document is to provide practical instructions to produce quality maps using open-source software. The author of this guide wants to make it clear that no quality maps can be produced if low quality inputs are used. Even the most sophisticated geostatistical tools will not be able to save the data sets of poor quality. A quality point data set is the one that fulfills the following requirements:

- *It is large enough* — The data set needs to be large enough to allow statistical testing. Typically, it is recommended to avoid using $\ll$50 points for reliable variogram modeling and $\ll$10 points per predictor for reliable regression modeling[2].

- *It is representative* — The data set needs to represent the area of interest, both considering the geographical coverage and the diversity of environmental features. In the case that parts of the area or certain environmental features (land cover/use types, geomorphological strata and similar) are misrepresented or completely ignored, they should be masked out or revisited.

- *It is independent* — The samples need to be collected using an objective sampling technique. The selection of locations needs to be done in an unbiased way so that no special preference is given to locations which are easier to visit, or are influenced by any other type of human bias. Preferably, the point locations should be selected using objective sampling designs such as simple random sampling, regular sampling, stratified random sampling or similar.

---

[2]Reliability of a variogram/regression model decreases exponentially as n approaches small numbers.

- *It is produced using a consistent methodology* — The field sampling and laboratory analysis methodology needs to be consistent, i.e. it needs to comprise standardized methods that are described in detail and therefore reproducible. Likewise, the measurements need to consistently report applicable support size and time reference.

- *Its precision is significantly precise* — Measurements of the environmental variables need to be obtained using field measurements that are significantly more precise than the natural variation.

Geostatistical mapping using inconsistent point samples[3], small data sets, or subjectively selected samples is also possible, but it can lead to many headaches — both during estimation of the spatial prediction models and during interpretation of the final maps. In addition, analysis of such data can lead to unreliable estimates of the model in parts or in the whole area of interest. As a rule of thumb, one should consider repetition of a mapping project if the prediction error of the output maps exceeds the total variance of the target variables in $\geq 50\%$ of the study area.

---

[3]Either inconsistent sampling methodology, inconsistent support size or inconsistent sampling designs.

Frequently Asked Questions

(1.) **Is spline interpolation different from kriging?**

In principle, splines and kriging are very similar techniques. Especially regularized splines with tension and universal kriging will yield very similar results. The biggest difference is that the splines require that a users sets the smoothing parameter, while in the case of kriging the smoothing is determined objectively. See also §1.2.3.

(2.) **What is experimental variogram and what does it shows?**

Experimental variogram is a plot showing how half of the squared differences between the sampled values (semivariance) changes with the distance between the point-pairs. We typically expect to see smaller semivariances at shorter distances and then a stable semivariance (equal to global variance) at longer distances. See also §1.3.1 and Fig. 1.7.

(3.) **How do I model anisotropy in a variogram?**

By adding two additional parameters — angle of the principal direction (strongest correlation) and the anisotropy ratio. You do not need to fit variograms in different directions. In gstat, you only have to indicate that there is anisotropy and the software will fit an appropriate model. See also Fig. 1.9.

(4.) **What is stationarity and should I worry about it?**

Stationarity is a property of a variable to have similar statistical properties (similar histogram, similar variogram) within the whole area of interest. There is the first-order stationarity or the stationarity of the mean value and the second-order stationarity or the covariance stationarity. The mean and covariance stationarity and a normal distribution of values are the requirements for ordinary kriging. In the case of regression-kriging, the target variable does not have to be stationary, but only its residuals. See also §1.3.1.

(5.) **What is the difference between regression-kriging, universal kriging and kriging with external drift?**

In theory, all three names describe the same technique. In practice, there are some computational differences: in the case of regression-kriging, the deterministic (regression) and stochastic (kriging) predictions are done separately; in the case of kriging with external drift, both components are fitted simultaneously; the term universal kriging is often reserved for the case when the deterministic part is modelled as a function of coordinates. See also §2.1.2.

(6.) **Can I interpolate categorical variables using regression-kriging?**

A categorical variable can be treated by using logistic regression (i.e. multinomial logistic regression if there are more categories). The residuals can then be interpolated using ordinary kriging and added back to the deterministic component. Ideally, one should use memberships $\mu \in (0, 1)$ which can be directly converted to logits and then treated as continuous variables. See also §2.3 and Figs. 4.14 and 4.19.

(7.) **How can I produce geostatistical simulations using a regression-kriging model?**

The gstat package allows users to generate multiple Sequential Gaussian Simulations using a regression-kriging model. However, this can be computationally demanding for large datasets. See also §2.4 and Fig. 1.2.

(8.) **How can I run regression-kriging on spatio-temporal point/raster data?**

You can extend the 2D space with time dimension if you simply treat it as the 3rd space- dimension. Then you can also fit 3D variograms and run regression models where observations are available in different time '*positions*'. Usually the biggest problem of spatio-temporal regression-kriging is to ensure enough ($\gg 10$) observations in time-domain. You also need to have time-series of predictors (e.g. time-series of remote sensing images). See also §2.5.

(9.) **Can co-kriging be combined with regression-kriging?**

Yes. Additional, more densely sampled covariates can be used to improve spatial interpolation of the residuals. The interpolated residuals can then be added to the deterministic part of variation.

(10.) **In which situations might regression-kriging perform poorly?**

Regression-kriging might perform poorly: if the point sample is small and nonrepresentative, if the relation between the target variable and predictors is non-linear, if the points do not represent feature space or represent only the central part of it. See also §2.8.2.

(11.) **In which software can I run regression-kriging?**

Regression-kriging, in full capacity, can be run in SAGA and gstat (implemented in R and Idrisi). SAGA has an user-friendly environment to enter the prediction parameters, however, it does not offer possibilities for more extensive statistical analysis (especially variogram modelling is very limited). R seems to be the most suitable computing environment for regression-kriging as it allows largest family of statistical methods and supports data processing automation. See also §3.7.1.

xiii

(12.) Can I run regression-kriging in ArcGIS?

In principle: No. In ArcGIS, as in ILWIS, it is possible to run separately regression and kriging of residuals and then sum the maps, but it does not support regression-kriging as explained in §2.1, nor simulations using a regression-kriging model. As any other GIS, ArcGIS has limits considering the sophistication of the geostatistical analysis. The statistical functionality of ArcView can be extended using the S-PLUS extension.

(13.) How do I export results of spatial prediction (raster maps) to Google Earth?

In ILWIS, you will first need to resample the map to the LatLonWGS84 system. Then you can export the map as a graphical file (BMP) and insert it into Google Earth as a ground overlay. You will need to know the bounding coordinates of the map expressed in geographic degrees. See also §3.5.2.

(14.) Why should I invest my time to learn R language?

R is, at the moment, the cheapest, the broadest, and the most professional statistical computing environment. In addition, it allows data processing automation, import/export to various platforms, extension of functionality and open exchange of scripts/packages. From few years ago, it also allows handling and generation of maps. The official motto of an R guru is: *anything is possible on R*!

(15.) What do I do if I get stuck with R commands?

Study the R Html help files, browse the R News, purchase the books on R, subscribe to the R mailing lists, obtain user-friendly R editors such as Tinn-R or use the package R commander (Rcmdr). The best way to learn R is to look at the existing scripts.

(16.) How can I handle large datasets ($\gg 10^3$ points, $\gg 10^6$ pixels) in R?

One option is to split the study area into regular blocks (e.g. 20 blocks) and then run the predictions separately for each block, but using the global model. You can also try installing/using some of the R packages develop to handle large datasets. See also §3.7.2.

(17.) How do I determine a suitable grid size for output maps?

The grid size of the output maps needs to match the sampling density and scale at which the processes of interest occur. We can always try to produce maps by using the most detail grid size that our predictors allow us. Then, we can slowly test how the prediction accuracy changes with coarser grid sizes and finally select a grid size that allows maximum detail, while being computationally effective. See also §4.2.3 and §4.7.1.

(18.) What is logit transformation and why should I use it?

Logit transformation converts the values bonded by two physical limits (e.g. min=0, max=100%) to $[-\infty, +\infty]$ range. It requires no intervention by user and it often helps improving the normality of the target variable (residuals), which is often a requirement for regression analysis. See also Fig. 4.15.

(19.) Why derive principal components of predictors (maps) instead of using the original predictors?

Principal Component Analysis is an useful technique to reduce overlap of information in the predictors. If combined with step-wise regression, it will typically help us determine the smallest possible subset of significant predictors. See also §4.2.4.

(20.) How do I set the right coordinate system in R?

By setting the parameters of the CRS argument of a spatial data frame. Obtain the European Petroleum Survey Group (EPSG) Geodetic Parameter database, and try to locate the exact CRS parameters by browsing the existing *Coordinate Reference System*. See also §4.2.1.

(21.) How can I evaluate the quality of my sampling plan?

For each existing point sample you can: evaluate clustering of the points by comparing the sampling plan with a random design, evaluate the representativity of sampling in both geographical and feature space (histogram comparison), evaluate consistency of the sampling intensity. See also §4.2.3.

(22.) How can I test if the difference between the two histograms of the same feature is significant?

Use a non-parametric test such as the Kolmogorov-Smirnov test. Both histograms must have the same intervals and show probability values. See also Fig. 4.9.

(23.) How do I set an initial variogram?

One possibility is to use: nugget parameter = measurement error, sill parameter = sampled variance, and range parameter = 10% of the spatial extent of the data (or two times the mean distance to the nearest neighbour). This is only an empirical formula. See also §4.4.

(24.) Can I automate regression-kriging so that no user-input is needed?

Automation of regression-kriging is possible in R. An user can combine data import, step-wise regression, variogram fitting, spatial prediction (gstat), and completely automate generation of maps. See also §3.7.4.

(25.) How do I test if the two prediction methods are significantly different?

Derive *RMSE* at validation points for both techniques and then test the difference between the distributions using the two sample t-test. See also §4.6.

(26.) Can we really produce quality maps with much less samples than we originally planned (is down-scaling possible with regression-kriging)?

If the correlation with the environmental predictors is very strong, you do not need as many point observations to produce quality maps. In such cases, the issue becomes more how to locate the samples so that the extrapolation in the feature space is minimized. See also §2.6 and §4.7.2.

(27.) **How can I allocate additional observations to improve the precision of a map?**

You can use the package spatstat and then run weighted point pattern randomization with the map of the normalized prediction variance as the weight map. This will produce a random design with the inspection density proportional to the value of the standardized prediction error. In the next iteration, precision of your map will gradually improve. See also Fig. 4.27.

(28.) **How do I export R plots directly to Google Earth?**

PNG of R plots can be exported as Google Earth overlays using the maptools package. The procedure is explained in §4.8.2.

1

Theoretical backgrounds

1.1 Basic concepts

Geostatistics is a subset of statistics specialized in analysis and interpretation of geographically referenced data (Goovaerts, 1997; Webster and Oliver, 2001; Nielsen and Wendroth, 2003). In other words, geostatistics comprises statistical techniques that are adjusted to spatial data. Typical questions of interest to a geostatistician are:

- *how does a variable vary in space?*

- *what controls its variation in space?*

- *where to locate samples to describe its spatial variability?*

- *how many samples are needed to represent its spatial variability?*

- *what is a value of a variable at some new location?*

- *what is the uncertainty of the estimate?*

In the most pragmatic context, geostatistics is an analytical tool for statistical analysis of sampled field data. Today, geostatistics is not only used to analyse point data but also increasingly in combination with various GIS layers: e.g. to explore spatial variation in remote sensing data, to quantify noise in the images and for their filtering (e.g. filling of the voids/missing pixels), to improve generation of DEMs and for their simulations, to optimize spatial sampling, selection of spatial resolution for image data and selection of support size for ground data (Kyriakidis et al., 1999; Atkinson and Quattrochi, 2000).

According to the bibliographic research of Zhou et al. (2007), the top 10 application fields of geostatistics (the largest number of research articles) are: (1) geosciences, (2) water resources, (3) environmental sciences, (4) agriculture and/or soil sciences, (5/6) mathematics and statistics, (7) ecology, (8) civil engineering, (9) petroleum engineering and (10) limnology. The list could be extended and differs from country to country of course. Evolution of applications of geostatistics can also be followed through the activities of the following research groups: geoENVia, IAMG, pedometrics, geocomputation and spatial accuracy.

One of the main uses of geostatistics is to predict values of a sampled variable over the whole area of interest, which is referred to as **spatial prediction** or **spatial interpolation**. Note that there is a small difference between the two because *prediction*

1

can imply both interpolation and extrapolation, so we will more commonly use the term *spatial prediction* in this handbook, even though the term *spatial interpolation* has been more widely accepted (Lam, 1983; Mitas and Mitasova, 1999; Dubois and Galmarini, 2004).

An important distinction between geostatistical and conventional mapping of environmental variables is that the geostatistical prediction is based on application of quantitative, statistical techniques. Unlike the traditional approaches to mapping, which rely on the use of empirical knowledge, in the case of **geostatistical mapping** we completely rely on the actual measurements and (semi-)automated algorithms. Although this sounds as if the spatial prediction is done purely by a computer program, the analysts have many options to choose whether to use linear or non-linear models, whether to consider spatial position or not, whether to transform or use the original data, whether to consider multicolinearity effects or not. So it is also an expert-based system in a way.

In summary, geostatistical mapping can be defined as **analytical production of maps by using field observations, auxiliary information and a computer program that calculates values at locations of interest** (a study area). It typically comprises the following five steps:

(1.) design the sampling and data processing,

(2.) collect field data and do laboratory analysis,

(3.) analyse the points data and estimate the model,

(4.) implement the model and evaluate its performance,

(5.) produce and distribute the output geoinformation[1].

Today, increasingly, the natural resource inventories need to be regularly updated or improved in detail, which means that after step (5), we often need to consider collection of new samples or additional samples that are then used to update an existing GIS layer. In that sense, it is probably more valid to speak about geostatistical **monitoring**.

1.1.1 Environmental variables

Environmental variables are quantitative or descriptive measures of different environmental features. Environmental variables can belong to different domains, ranging from biology (distribution of species and biodiversity measures), soil science (soil properties and types), vegetation science (plant species and communities, land cover types), climatology (climatic variables at surface and benith/above), hydrology (water quantities and conditions) and similar. They are commonly collected through field sampling (supported by remote sensing), which are then used to produce maps showing their distribution in an area. Such accurate and up-to-date maps of environmental features represent a crucial input to spatial planning, decision making, land evaluation or land degradation assessment (Table 1.1).

From a meta-physical perspective, what we are most often mapping in geostatistics are, in fact, quantities of molecules of a certain kind. For example, a measure of soil or water acidity is the pH factor. By definition, pH is a negative exponent of the

---

[1] By this I mainly think of on-line databases, i.e. data distribution portals.
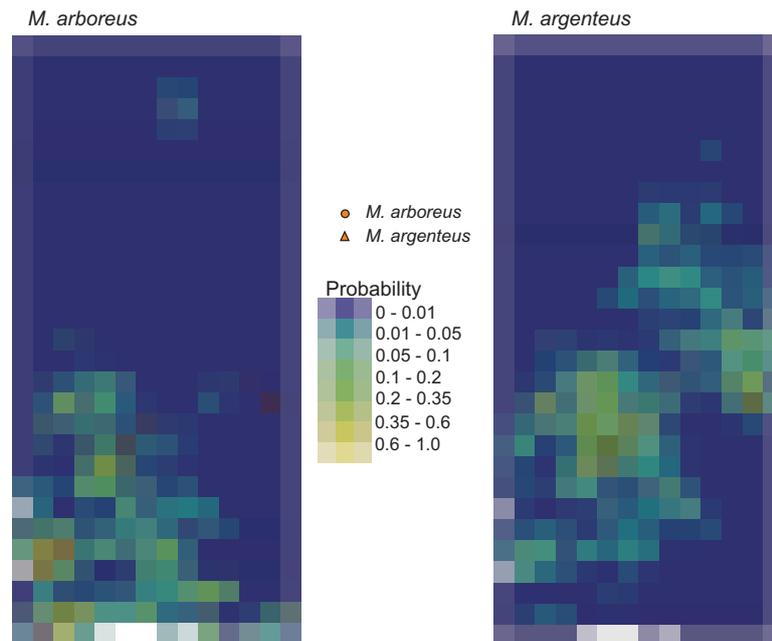
Fig. 1.1: Example of a geostatistical mapping of occurrence of sister (plant) species. After Latimer et al. (2004).

concentration of the H^+ ions[2]. By mapping it over the whole area of interest, we produce a map of continuous values of concentration (**continuous fields**) of H^+ ions.

In the case of plants and animals, geostatistical mapping becomes much more complicated. Here, we deal with distinct physical **objects** (individuals), often immeasurable in quantity. In addition, animal species change their location dynamically, often in unpredictable directions and with unpredictable spatial patterns (non-linear trajectories), which asks for high sampling density in both space and time domains (Table 1.1).

To account for these problems, spatial modelers rarely aim at mapping the distribution of individuals (e.g. represented as points), but instead use compound measures that are suitable for management and decision making purposes. For example, animal species can be represented using density or biomass measures (see e.g. Latimer et al. (2004); Pebesma et al. (2005)). In vegetation mapping, most commonly field observations of the plant **occurrence** (ranging from 0 to 100%) are recorded (Fig. 1.1). In addition to mapping of temporary distribution of species, biologist aim at developing statistical models to define optimal ecological conditions for a certain species. This is often referred to as **habitat mapping** (Latimer et al., 2004; Antonić et al., 2005) and can be also dealt with geostatistics. Occurrence of species or habitat conditions can also be presented as continuous fields, i.e. using raster maps.

1.1.2 Aspects of spatial variability

Relevant and detailed geoinformation is a prerequisite for successful management of natural resources in many applied environmental and geosciences. Until recently, maps

---

[2]It is often important to understand the definition of an environmental variable. For example, in the case of pH, we should know that the quantities are already on a log-scale so that no further transformation of the variable is anticipated (see further §4.2.1).

Table 1.1: Some common environmental variables of interest to decision making and their properties: SRV — short-range variability; TV — temporal variability; VV — vertical variability; SSD — standard sampling density; DRS — remote-sensing detectability. ★ — high, ⋆ — medium, − — low or non-existent. Approximated by the author.

Environmental features/topics	Common variables of interest to decision making	SRV	TV	VV	SSD	RSD
Mineral exploration: oil, gas, mineral resources	mineral occurrence and concentrations of minerals; reserves of oil and natural gas; magnetic anomalies;	⋆	−	★	⋆	⋆
Freshwater resources and water quality	O_2, ammonium and phosphorus concentrations in water; concentration of herbicides; trends in concentrations of pollutants; temperature change;	⋆	⋆	⋆	⋆	−
Socio-economic parameters	population density; population growth; GDP per km^2; life expectancy rates; human development index; noise intensity;	⋆	⋆	−	★	★
Land degradation: erosion, landslides, surface runoff	soil loss; erosion risk; quantities of runoff; dissolution rates of various chemicals; landslide susceptibility;	⋆	⋆	−	−	★
Natural hazards: fires, floods, earthquakes, oil spills	burnt areas; fire frequency; water level; earthquake hazard; financial losses; human casualties; wildlife casualties;	★	★	−	⋆	★
Human-induced radioactive contamination	gama doze rates; concentrations of isotopes; PCB levels found in human blood; cancer rates;	⋆	★	−	⋆	★
Soil fertility and productivity	organic matter, nitrogen, phosphorus and potassium in soil; biomass production; (grain) yields; number of cattle per ha; leaf area index;	★	⋆	⋆	⋆	⋆
Soil pollution	concentrations of heavy metals especially: arsenic, cadmium, chromium, copper, mercury, nickel, lead and hexachlorobenzene; soil acidity;	★	⋆	−	★	−
Distribution of animal species (wildlife)	occurrence of species; biomass; animal species density; biodiversity indices; habitat conditions;	★	★	−	⋆	−
Distribution of natural vegetation	land cover type; vegetation communities; occurrence of species; biomass; density measures; vegetation indices; species richness; habitat conditions;	⋆	⋆	−	★	★
Meteorological conditions	temperature; rainfall; albedo; cloud fraction; snow cover; radiation fluxes; net radiation; evapotranspiration;	⋆	★	⋆	⋆	★
Climatic conditions and changes	mean, minimum and maximum temperature; monthly rainfall; wind speed and direction; number of clear days; total incoming radiation; trends of changes of climatic variables;	−	★	⋆	⋆	⋆
Global atmospheric conditions	aerosol size; cirrus reflectance; carbon monoxide; total ozone; UV exposure;	⋆	★	★	−	★
Air quality in urban areas	NO_x, SO_2 concentrations; emission of greenhouse gasses; emission of primary and secondary particles; ozone concentrations; Air Quality Index;	★	★	★	★	−
Global and local sea conditions	chlorophyll concentrations; biomass; sea surface temperature; emissions to sea;	⋆	★	⋆	⋆	⋆

of environmental variables have primarily generated by using mental models (expert systems). Because field data collection is often the most expensive part of a survey, survey teams typically visit only a limited number of sampling locations and then, based on the sampled data and statistical and/or mental models, infer conditions for the whole area of interest. As a consequence, maps of environmental variables have often been of limited and inconsistent quality and usually too subjective.

Spatial variability of environmental variables is commonly a result of complex processes working at the same time and over long periods of time, rather than an effect of a single realization of a single factor. To explain variation of environmental variables has never been an easy task. Many environmental variables vary not only horizontally but also with depth, not only continuously but also abruptly. Field observations are, on the other hand, usually very expensive and we are often forced to build 100% complete maps by using a sample of $\ll 1\%$.
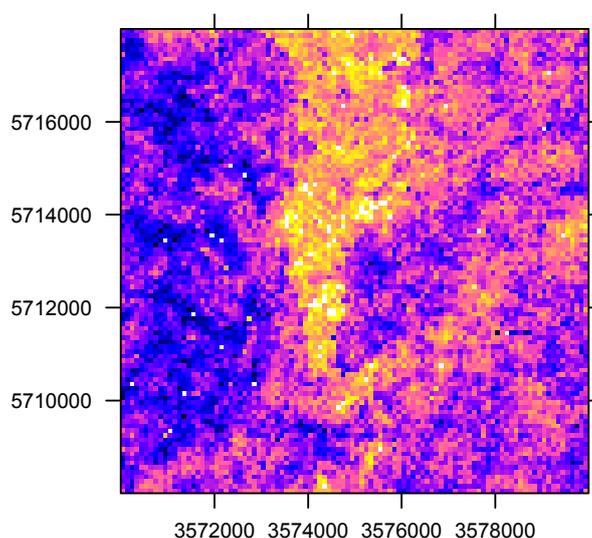
Fig. 1.2: If we were able to sample a soil variable over the whole area of interest, we would probably get an image such as this. This image was, in fact, produced using the geostatistical simulations with a regression-kriging model (see further §4.18).

Imagine if we had enough funds to inventory each grid node in a study area, then we would be able to produce a map which would probably look as the map shown in Fig. 1.2[3]. By carefully looking at this map, you can notice several things: (1) there seems to be a spatial pattern of how the values change; (2) values that are closer together are more similar; (3) locally, the values can differ without any systematic rule (randomly); (4) in the middle of the area, the values seem to be in general higher (a discrete change) etc. From the statistical perspective, an environmental variable can be viewed as an information *signal* consisting of three components:

$$Z(\mathbf{s}) = Z^*(\mathbf{s}) + \varepsilon'(\mathbf{s}) + \varepsilon'' \qquad (1.1.1)$$

where $Z^*(\mathbf{s})$ is the deterministic component, $\varepsilon'(\mathbf{s})$ is the spatially correlated random component and ε'' is the pure noise, usually the result of the measurement error. This model is in literature often referred to as the **universal model of variation** (see further §2.1). Note that we use a capital letter Z because we assume that the model

---

[3]See further §4.5 and Fig. 2.1.

is probabilistic, i.e. there is a range of equiprobable realisations of the same model $\{Z(\mathbf{s}), \mathbf{s} \in \mathbb{A}\}$.

In theory, we could decompose a map of an environmental variable into two grids: (1) the deterministic and (2) the error surface; in practice, we are not able to do distinguish the deterministic from the error part of the signal because both can show similar patterns. By collecting field measurements at different locations and with different sampling densities, we might be able to infer about the source of variability and estimate probabilistic models of variation. Then we can try to answer how much of the variation is due to the measurement error, how much has been accounted for by the environmental factors, how much is due to the spatial similarity of the values and how much is uncorrelated noise? Such systematic assessment of the error budget allows us to make realistic interpretations and utilize models which reflect our knowledge about the variability of target variables.

The first step towards successful geostatistical mapping of environmental variables is to understand the sources of variability in the data. As we have seen previously, the variability is a result of deterministic and stochastic processes plus the pure noise. In other words, the variability in data is a sum of two components: (a) the **natural spatial variation** and (b) the **inherent noise**, mainly do the measurement errors (Burrough and McDonnell, 1998). Measurement errors typically occur during the positioning in the field, during sampling or the laboratory analysis. These errors should ideally be minimized, because they are not of primary concern for a mapper. What the mappers are interested in is the natural spatial variation, which is mainly due to the physical processes that can be explained (up to a certain level) by a mathematical model.
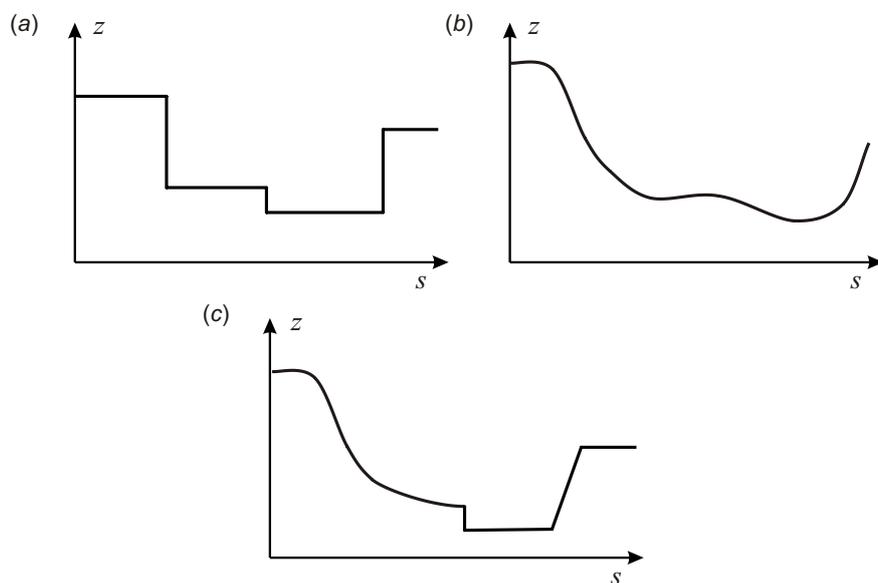
Fig. 1.3: Schematic examples of models of spatial variation: abrupt changes of values can be modelled using a discrete model of spatial variation (a), smooth changes can be modelled using a continuous model of spatial variation (b). In reality, we often need to work with a mixed (or hybrid) model of spatial variation (c).

The second step towards reliable modelling is to consider all aspects of natural variation. Although spatial prediction of environmental variables is primarily concerned with *geographical* variability, there are also other aspects of natural soil variation that are often overlooked by mappers: *vertical*, *temporal* and *scale* aspect. Below is an overview

of the main concepts and problems associated with each of these (see also Table 1.1):

Geographical variation (2D) The results of spatial prediction are either visualized as 2D maps or cross-sections. Some environmental variables, such as thickness of soil horizons, the occurrence of vegetation species or soil types, do not have a third dimension, i.e. they refer to the Earth's surface only. Others, such as temperature, population densities etc. can be measured at various altitudes, even below Earth's surface. Geographical part of variation can be modelled using either a **continuous**, **discrete** or **mixed model of spatial variation** (Fig. 1.3).

Vertical variation (3D) Many environmental variables also vary with depth or altitude. In many cases, the measured difference between the values is higher at a depth differing by a few centimetres than at geographical distance of few meters. Consider variables such as temperature or bird density — to explain their vertical distribution can often be more difficult than for the horizontal space (Shamoun et al., 2005). Transition between different soil layers, for example, can also be both gradual and abrupt, which requires a double-mixed model of soil variation for 3D spatial prediction. Some authors suggest the use of cumulative values on volume (areal) basis to simplify mapping of the 3D variables. For example, McKenzie and Ryan (1999) produced maps of total phosphorus and carbon estimated in the upper 1 m of soil and expressed in tons per hectare, which then simplifies production and retrieval.

Temporal variation As mentioned previously, especially environmental variables connected with animal and plant species vary not only within season but often within few moments. Even the soil variables such as pH, nutrients, water-saturation levels and water content, can vary over a few years, within a single season or even over a few days (Heuvelink and Webster, 2001). Temporal variability makes geostatistical mapping especially complex and expensive. Maps of environmental variables produced for two different time references can differ significantly. This means that most of maps are valid for a certain period (or moment) of time only. In many cases the seasonal periodicity of environmental variables is regular, so that prediction does not necessarily require new samples (see further §2.5).
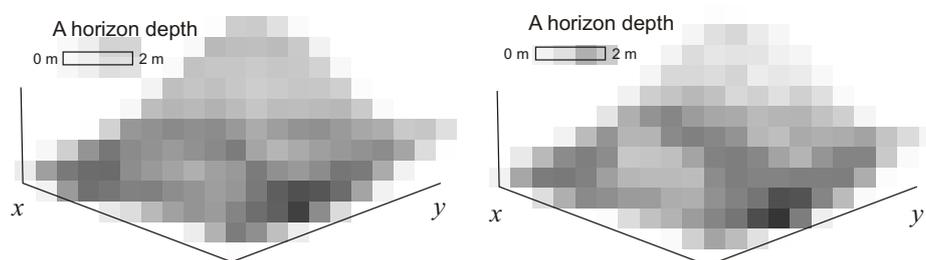
Fig. 1.4: Influence of the support (grid cell) size: predictions of the same variable at coarse grid will often show much less contrast. Example from Thompson et al. (2001).

Support size Support size is the discretisation level of a geographical surface and is related to the concept of scale. In the case of spatial predictions, there are two support sizes: the size of the blocks of land sampled, and grid resolution of the auxiliary maps. Field observations are typically collected as point samples. The

support size of the auxiliary maps is commonly much larger than the actual blocks of land sampled, e.g. auxiliary variables are in general averaged (smoothed), while the environmental variables can describe local (micro) features. As a result, the correlation between the auxiliary maps and measured environmental variables is often low or insignificant (Fig. 1.4). There are two solutions to this problem: (a) to up-scale the auxiliary maps or work with super-high resolution/detail data (e.g. IKONOS images of 1 m resolution), or (b) to average bulk or composite samples within the regular blocks of land (Patil, 2002). The first approach is more attractive for the efficiency of prediction, but at the cost of more processing power and storage. The second solution will only result in a better fit, whereas the efficiency of prediction, validated using point observations, may not change significantly.

This means that mixing of lab data from different seasons, depths and with different support sizes in general means lower predictive power and problems in fully interpreting the results. If the focus of prediction modelling is solely the geographical component (2D), then the samples need to be taken under fixed conditions: same season, same depths, same blocks of land. This also means that each 2D map of an environmental variable **should always indicate a time reference (interval), applicable vertical dimension**[4] **and the sample (support) size i.e. the effective scale**.

1.1.3   Spatial prediction models

Ideally, variability of environmental variables is determined by a finite set of inputs and they exactly follow some known physical law. If the algorithm (formula) is known, the values of the target variables can be predicted exactly. In reality, the relationship between the feature of interest and physical environment is so complex[5] that it cannot be modelled exactly (Heuvelink and Webster, 2001). This is because we either do not exactly know: (a) the final list of inputs into the model, (b) the rules (formulas) required to derive the output from the inputs and (c) the significance of the random component in the system. So the only possibility is that we can try to estimate a model by using the actual field measurements of the target variable. This can be referred to as the *indirect* or *non-deterministic* estimation.

Let us first define the problem using mathematical notation. Let a set of observations of a **target variable** z be denoted as $z(\mathbf{s}_1)$, $z(\mathbf{s}_2)$,..., $z(\mathbf{s}_n)$, where $\mathbf{s}_i = (x_i, y_i)$ is a location and x_i and y_i are the coordinates (primary locations) in geographical space and n is the number of observations (Fig. 1.5). The geographical domain of interest (area, land surface, object) can be denoted as $\mathbb{A}$. Assuming that the samples are *representative*, *unbiased* and *consistent* (see further §4.2.3), values of the target variable at some new location $\mathbf{s}_0$ can be derived using a **spatial prediction model**. It defines inputs, outputs and the computational procedure to derive outputs based on the given inputs:

$$\hat{z}(\mathbf{s}_0) = E\left\{Z | z(\mathbf{s}_i),\ q_k(\mathbf{s}_0),\ \gamma(\mathbf{h}),\ \mathbf{s} \in \mathbb{A}\right\} \tag{1.1.2}$$

where $z(\mathbf{s}_i)$ is the input point dataset, $q_k(\mathbf{s}_0)$ is the list of deterministic predictors and $\gamma(\mathbf{h})$ is the covariance model defining the spatial autocorrelation structure (see further Fig. 2.1).

---

[4]Orthogonal distance from the land surface.

[5]Because either the factors are unknown, or they are too difficult to measure, or the model itself would be too complex for realistic computations.
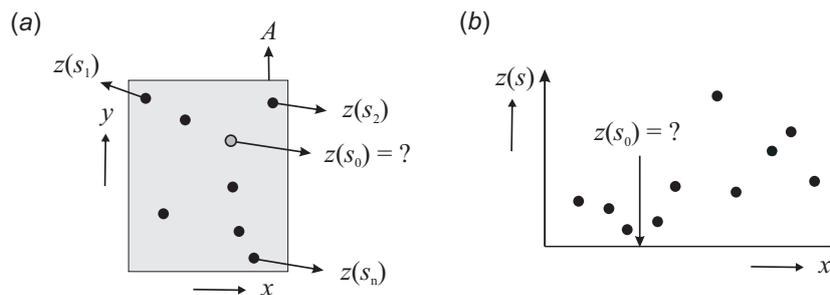
Fig. 1.5: Spatial prediction is a process of estimating the value of (quantitative) properties at unvisited site within the area covered by existing observations: (a) a scheme in horizontal space, (b) values of some target variable in a one-dimensional space.

In raster GIS terms, the geographical domain of interest is a rectangular matrix, i.e. an array with a (large) number of grid nodes over the domain of interest (Fig. 1.6):

$$\mathbf{Z} = \{Z(\mathbf{s}_j), j = 1, \ldots, N\}; \quad \mathbf{s}_j \in \mathbb{A} \tag{1.1.3}$$

where $\mathbf{Z}$ is the data array, $Z(\mathbf{s}_j)$ is the value at the grid node $\mathbf{s}_j$, and m is the total number of grid nodes. Note that there is a difference between predicting values at grid node (punctual) and prediction values of the whole grid cell (block), which has a full topology[6].

There seems to be many possibilities to interpolate point samples. At the Spatial Interpolation Comparison 2004 exercise, for example, 31 algorithms competed in predicting values of gamma dose rates at 1008 new locations and by using 200 training data (Dubois and Galmarini, 2004; Dubois, 2005). The competitors ranged from splines, neural networks up to various kriging algorithms. Similarly, the software package Surfer offers dozens of interpolation techniques: Inverse Distance, Kriging, Minimum Curvature, Polynomial Regression, Triangulation, Nearest Neighbour, Shepard's Method, Radial Basis Functions, Natural Neighbour, Moving Average, Local Polynomial, etc.
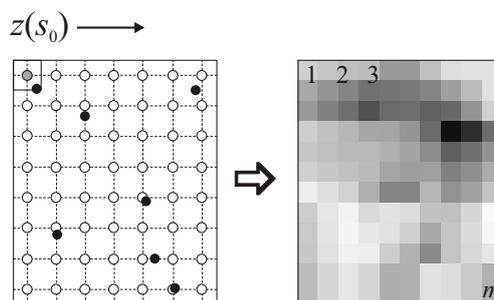
Fig. 1.6: Spatial prediction implies application of a prediction algorithm to an array of grid nodes (*point á point* spatial prediction). The results are then displayed using a raster map.

An inexperienced user will often be confused by amount of techniques. Most of the spatial prediction models are in fact somehow connected. As we will see later on, many

---

[6]The sp package in R, for example, makes a distinction between the Spatial Pixel data frame (grid nodes) and a Spatial Grid data frame (grid cells).

models are in fact just a special case of a more general prediction model. This makes things much less complicated for the non-geostatisticians[7].

Spatial prediction models (algorithms) can be classified based on several aspects. Most importantly, they can be classified according to the amount of statistical analysis included:

MECHANICAL/EMPIRICAL MODELS — These are models where arbitrary or empirical model parameters are used. No estimate of the model error is available and usually no strict assumptions about the variability of a feature exist. The most known techniques that belong to this group are:

- *Thiessen polygons*;
- *Inverse distance interpolation*;
- *Regression on coordinates*;
- *Splines*;
- . . .

STATISTICAL (PROBABILITY) MODELS — In the case of statistical models, the model parameters are commonly estimated in an objective way, following the probability theory. The predictions are accompanied with the estimate of the prediction error. A drawback is that the input dataset usually need to satisfy strict statistical assumptions. There are at least four groups of statistical models:

- *kriging* (plain geostatistics);
- *environmental correlation* (e.g. regression-based);
- *Bayesian-based models* (e.g. Bayesian Maximum Entropy);
- *mixed models* (regression-kriging);
- . . .

Spatial prediction models can also be grouped based on the:

Smoothing effect — whether the model smooths predictions at sampling locations or not:

- *Exact* (measured and estimated values coincide);
- *Approximate* (measured and estimated values do not have to coincide);

Proximity effect — whether the model uses all sampling locations or only locations in local proximity:

- *Local* (a local sub-sample; local models applicable);
- *Global* (all samples; the same model for the whole area);

Convexity effect — whether the model makes predictions outside range of the data:

- *Convex* (all predictions are within the range);
- *Non-convex* (some predictions might be outside the range);

Support size — whether the model predicts at points or for blocks of land:

---

[7]As we will see later on in §2.1.1, spatial prediction can even be fully automated so that a user needs only to provide quality inputs and the system will select the most suitable technique.

- *Point-based* or punctual prediction models;
- *Area-based* or block prediction models;

Another way to look at the spatial prediction models is their ability to represent models of spatial variation. Ideally, we wish to use mixed model of spatial variation (Fig. 1.3c) because it is a generalization of the two models and can be more universally applied. In practice, many spatial prediction models are limited to one of the two models of spatial variation: predicting using polygon maps (§1.3.3) will show discrete changes (Fig. 1.3a) in values; ordinary kriging (§1.3.1) will typically lead to smooth maps (Fig. 1.3b).

1.2 Mechanical spatial prediction models

As mentioned previously, mechanical spatial prediction models can be very flexible and easy to use. They can be considered to be subjective or empirical techniques because the user him/her-self selects the parameters of the model, often without any deeper statistical analysis. Most commonly, a user typically accepts the default parameters suggested by some software, hence the name *mechanical* models. The most used mechanical spatial prediction models are Thiessen polygons, inverse distance interpolation, regression on coordinates and splines, although the list could be extended (Lam, 1983; Myers, 1994). In general, mechanical prediction models are more primitive than the statistical models and often sub-optimal, however, there are situations where they can perform as good as the statistical models (or better).

1.2.1 Inverse distance interpolation

Probably one of the oldest spatial prediction technique is the **inverse distance interpolation** (Shepard, 1968). As with many other spatial predictors, in the case of the inverse distance interpolation, a value of target variable at some new location can be derived as a weighted average:

$$\hat{z}(\mathbf{s}_0) = \sum_{i=1}^{n} \lambda_i(\mathbf{s}_0) \cdot z(\mathbf{s}_i) \tag{1.2.1}$$

where λ_i is the weight for neighbour i. The sum of weights needs to equal one to ensure an unbiased interpolator. Eq.(1.2.1) in matrix form is:

$$\hat{z}(\mathbf{s}_0) = \lambda_{\mathbf{0}}^{\mathbf{T}} \cdot \mathbf{z} \tag{1.2.2}$$

The simplest approach for determining the weights is to use the **inverse distances** from all points to the new point:

$$\lambda_i(\mathbf{s}_0) = \frac{\frac{1}{d^\beta(\mathbf{s}_0, \mathbf{s}_i)}}{\sum\limits_{i=0}^{n} \frac{1}{d^\beta(\mathbf{s}_0, \mathbf{s}_i)}}; \qquad \beta > 1 \tag{1.2.3}$$

where $d(\mathbf{s}_0, \mathbf{s}_i)$ is the distance from the new point to a known sampled point and β is a coefficient that is used to adjust the weights. This way, points which are close to an output pixel will obtain large weights and that points which are farther away from an output pixel will obtain small weights. The higher the β, the less importance will be put on distant points. The remaining problem is how to estimate β objectively so that it reflects the inherent properties of a dataset.

Inverse distance interpolation is an exact and convex interpolation method that fits only the continuous model of spatial variation. For large datasets ($\gg 10^3$ points) it can be time-consuming so it is often a good idea to set a threshold distance (search radius) to speed up the calculations.

1.2.2 Regression on coordinates

Assuming that the values of target variable at some location are function of coordinates, we can determine its values by finding a function which passes through (or close to) the given set of discrete points. This group of techniques can be termed *regression on coordinates*, although it is primarily known in literature by names **trend surfaces** and/or **moving surface interpolation**, depending on whether the function is fitted for the whole point dataset (trend) or for a local (moving) neighbourhood (Hardy, 1971). Regression on coordinates is based on the following model (Webster and Oliver, 2001, p.40–42):

$$Z(\mathbf{s}) = f(x, y) + \varepsilon \tag{1.2.4}$$

and the predictions are made by:

$$\hat{z}(\mathbf{s_0}) = \sum_{r,s \in n} a_{rs} \cdot x^r y^s = \mathbf{a^T} \cdot \mathbf{s_0} \tag{1.2.5}$$

where $r + s < p$ is the number of transformations of coordinates, p is the order of the surface. The model coefficients ($\mathbf{a}$) are determined by maximising the local fit:

$$\sum_{i=1}^{n} (\hat{z}_i - z_i)^2 \to \min \tag{1.2.6}$$

which can be achieved by the **Ordinary Least Squares** solution (Neter et al., 1996):

$$\mathbf{a} = \left(\mathbf{s^T} \cdot \mathbf{s}\right)^{-1} \cdot \left(\mathbf{s^T} \cdot \mathbf{z}\right) \tag{1.2.7}$$

In practice, local fitting of the moving surface is more used to generate maps than trend surface interpolation. In the case of the moving surface, for each output grid node, a polynomial surface is fitted to a larger[8] number of points selected by a moving window (circle). The main problem of this technique is that, by introducing higher order polynomials, we can generate many artefacts and cause serious overshooting of the values locally (see further Fig. 1.11). Moving surface will also completely fail to represent discrete changes in space.

Regression on coordinates can be criticized for not relying on empirical knowledge about the variation of a variable. As we will see later on in §1.3.2, it is probably more advisable to use feature-related **geographic predictors** such as the distance from a coast line, latitude or longitude and similar, instead of mechanically using the x, y coordinates and their transforms. In that sense, regression on coordinates[9] can be considered as the least sophisticated spatial prediction technique.

---

[8]The number of points need to be larger than the number of parameters.

[9]Similar can be said also for the Universal kriging where coordinates are used to explain the deterministic part of variation.

1.2.3 Splines

A special group of interpolation techniques is based on **splines**. A spline is a special type of piecewise polynomial and are preferable to simple polynomial interpolation because more parameters can be defined including the amount of smoothing. The smoothing spline function also assumes that there is a (measurement) error in the data that needs to be smoothed locally. There are many versions and modifications of spline interpolators. The most widely used techniques are **thin-plate splines** (Hutchinson, 1995) and **regularized spline with tension and smoothing** (Mitášová and Mitas, 1993).

In the case of regularized spline with tension and smoothing (implemented in GRASS GIS), the predictions are obtained by (Mitasova et al., 2005):

$$\hat{z}(\mathbf{s}_0) = a_1 + \sum_{i=1}^{n} w_i \cdot R(v_i) \tag{1.2.8}$$

where the a_1 is a constant and $R(v_i)$ is the radial basis function determined using (Mitášová and Mitas, 1993):

$$R(v_i) = -[E_1(v_i) + \ln(v_i) + C_E] \tag{1.2.9}$$

$$v_i = \left[\varphi \cdot \frac{\mathbf{h_0}}{2} \right]^2 \tag{1.2.10}$$

where $E_1(v_i)$ is the exponential integral function, $C_E=0.577215$ is the Euler constant, φ is the generalized tension parameter and $\mathbf{h_0}$ is the distance between the new and interpolation point. The coefficients a_1 and w_i are obtained by solving the system:

$$\sum_{i=1}^{n} w_i = 0 \tag{1.2.11}$$

$$a_1 + \sum_{i=1}^{n} w_i \cdot \left[R(v_i) + \delta_{ij} \cdot \frac{\varpi_0}{\varpi_i} \right] = z(\mathbf{s}_i); \qquad j = 1, .., n \tag{1.2.12}$$

where ϖ_0/ϖ_i are positive weighting factors representing a smoothing parameter at each given point $\mathbf{s}_i$. The tension parameter φ controls the distance over which the given points influence the resulting surface, while smoothing parameter controls the vertical deviation of of the surface from the points. By using an appropriate combination of tension and smoothing, one can produce a surface which accurately fits the empirical knowledge about the expected variation (Mitasova et al., 2005). Regularized spline with tension and smoothing are, in a way, equivalent to universal kriging (see further §2.1.2) where coordinates are used to explain the deterministic part of variation, and would yield very similar results.

Splines have shown to be highly suitable for interpolation of densely sampled heights and climatic variables (Hutchinson, 1995; Mitas and Mitasova, 1999). However, their biggest criticism is inability to incorporate larger amounts of auxiliary maps to model the deterministic part of variation. In addition, the smoothing and tension parameters need to be set by the user.

1.3 Statistical spatial prediction models

In the case of statistical models, coefficients/rules used to derive outputs are derived in an objective way following the theory of probability. Unlike mechanical models, in the

case of statistical models, we need to follow several statistical data analysis steps before we can generate maps. This makes the whole mapping process more complicated but it eventually helps us: (a) produce more reliable/objective maps, (b) understand the sources of errors in the data and (c) depict problematic areas/points that need to be revisited.

1.3.1 Kriging

Kriging has for many decades been used as a synonym for geostatistical interpolation. It originated in the mining industry in the early 1950's as a means of improving ore reserve estimation. The original idea came from the mining engineers D. G. Krige and the statistician H. S. Sichel. The technique was first published in Krige (1951), but it took almost a decade until a French mathematician G. Matheron derived the formulas and basically established the whole field of linear geostatistics[10] (Cressie, 1990; Webster and Oliver, 2001; Zhou et al., 2007).

A standard version of kriging is called **ordinary kriging** (**OK**). Here the predictions are based on the model:

$$Z(\mathbf{s}) = \mu + \varepsilon'(\mathbf{s}) \tag{1.3.1}$$

where μ is the constant *stationary* function (global mean) and $\varepsilon'(\mathbf{s})$ is the spatially correlated stochastic part of variation. The predictions are made as in Eq.(1.2.1):

$$\hat{z}_{\texttt{OK}}(\mathbf{s}_0) = \sum_{i=1}^{n} w_i(\mathbf{s}_0) \cdot z(\mathbf{s}_i) = \lambda_{\mathbf{0}}^{\mathbf{T}} \cdot \mathbf{z} \tag{1.3.2}$$

where $\lambda_{\mathbf{0}}$ is the vector of kriging weights (w_i), $\mathbf{z}$ is the vector of n observations at primary locations. In a way, kriging can be seen as a sophistication of the inverse distance interpolation. Recall from §1.2.1 that the key problem of inverse distance interpolation is to determine how much importance should be given to each neighbour. Intuitively thinking, there should be a way to estimate the weights in an objective way, so the weights reflect the true spatial autocorrelation structure. The novelty that Matheron (1962) and Gandin (1963) introduced to the analysis of point data is the derivation and plotting of the so-called **semivariances** — differences between the neighbouring values:

$$\gamma(\mathbf{h}) = \frac{1}{2} E \left[(z(\mathbf{s}_i) - z(\mathbf{s}_i + \mathbf{h}))^2 \right] \tag{1.3.3}$$

where $z(\mathbf{s}_i)$ is the value of target variable at some sampled location and $z(\mathbf{s}_i + \mathbf{h})$ is the value of the neighbour at distance $\mathbf{s}_i + \mathbf{h}$. Suppose that there are n point observations, this yields $n \cdot (n-1)/2$ pairs for which a semivariance can be calculated. We can then plot all semivariances versus their distances, which will produce a variogram cloud as shown in Fig. 1.7b. Such clouds are not easy to describe visually, so the values are commonly averaged for standard distance called the **lag**. If we display such averaged data, then we get a standard **experimental variogram** as shown in Fig. 1.7c. What we usually expect to see is that semivariances are smaller at shorter distance and then they stabilize at some distance. This can be interpreted as follows: the values of a target variable are more similar at shorter distance, up to a certain distance where the

---

[10]Matheron (1962) named his theoretical framework the *Theory of Regionalized Variables*. It was basically a theory for modelling stochastic surfaces using spatially sampled variables.

differences between the pairs are more less equal the global variance[11]. This is known as the *spatial auto-correlation effect.*
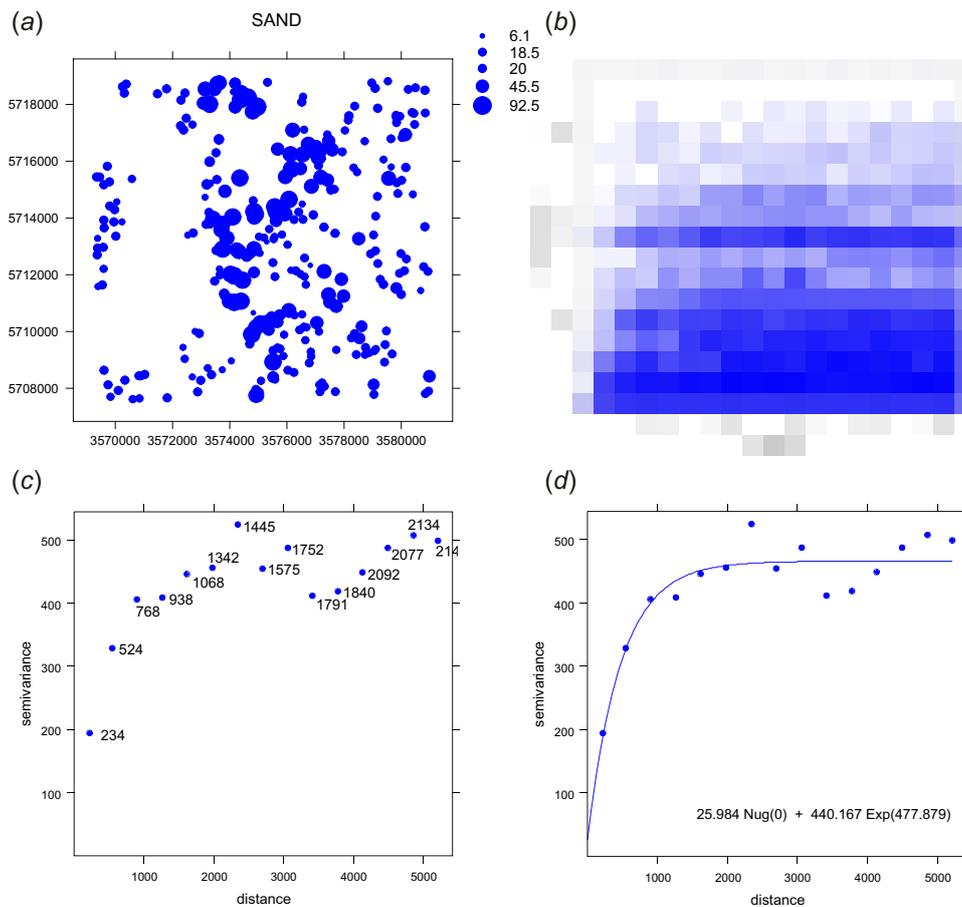
Fig. 1.7: Steps of variogram modelling: (a) location of points (300), (b) variogram cloud showing semivariances for 44850 pairs, (c) semivariances aggregated to lags of about 300 m, and (d) the final variogram model fitted using the default settings in `gstat`.

From a meta-physical perspective, spatial auto-correlation in the data can be considered as a result of **diffusion** — a random motion causing a system to decay towards uniform conditions. One can argue that, if there is a physical process behind a feature, one should model it using a deterministic function rather than to treat it as a stochastic component. However, diffusion is a random motion so that there is a meta-statistical argument to treat it is a stochastic component.

Once we calculated an experimental variogram, we can fit it using some of the **authorized variogram models**, such as *linear, spherical, exponential, circular, Gaussian, Bessel, power* and similar (Isaaks and Srivastava, 1989; Goovaerts, 1997). The variograms are commonly fitted by iterative reweighted least squares estimation, where the weights are determined based on the number of point pairs or based on the distance. Most commonly, the weights are determined using $N_j/\mathbf{h}_j^2$, where N_j is the number of pairs at certain lag, and $\mathbf{h}_j$ is the distance (Fig. 1.7d). This means that the algorithm will give much more importance to semivariances with large number of point pairs and to the shorter distances. Fig. 1.7d shows the result of automated variogram fitting given

---

[11]For this reason, many geostatistical packages (e.g. Isatis) automatically plot the global variance (horizontal line) directly in a variogram plot.

an experimental variogram (Fig. 1.7c) and using the $N_j/\mathbf{h}_j^2$-weights: in this case, we obtained an exponential model with the nugget parameter = 26, sill parameter = 440, and the range parameter = 478 m. Note that this is only a **sample variogram** — if we would go and collect several point samples, each would lead to somewhat different variogram plot. The target variable is said to be *stationary* if several sample variograms are very similar (constant), which is referred to as the **covariance stationarity**. Otherwise, if the variograms differ much locally and/or globally, then we speak about a non-stationary inherent properties. In principle, assumptions of kriging are that the target variable is stationary and that it has a normal distribution, which is probably the biggest limitation of kriging[12]. It is also important to note that there is a difference between the range factor and the range of spatial dependence, also known as the **practical range**. A practical range is the Lag $\mathbf{h}$ for which $\gamma(\mathbf{h})=0.95\ \gamma(\infty)$, i.e. that distance at which the semivariance is close to 95% of the sill (Fig. 1.8b).
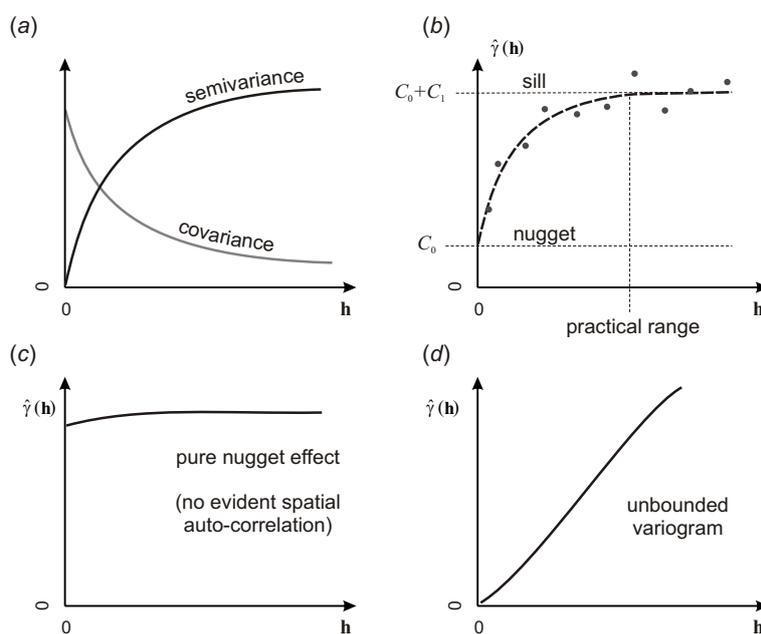
Fig. 1.8: Some basic concepts of variograms: (a) the difference between semivariance and covariance; (b) it often important in geostatistics to distinguish between the sill variation $(C_0 + C_1)$ and the sill parameter (C_1) and between the range parameter (R) and the practical range; (c) a variogram that shows no spatial correlation can be defined by a single parameter (C_0); (d) an unbounded variogram typically leads to predictions similar to inverse distance interpolation.

Once we have estimated[13] the variogram model, we can use it to derive semivariances at all locations and solve the kriging weights. The kriging OK weights are solved by multiplying the covariances:

$$\lambda_0 = \mathbf{C}^{-1} \cdot \mathbf{c_0}; \qquad C(|\mathbf{h}| = 0) = C_0 + C_1 \tag{1.3.4}$$

where $\mathbf{C}$ is the covariance matrix derived for $n \times n$ observations and $\mathbf{c_0}$ is the vector of

---

[12]The constant variogram/histogram and normality are rarely tested in real case studies, which can lead to poor predictions (although the output maps might appear to be fine). In the case of regression-kriging (see further §2.1), the variable does not have to be stationary, so no need to test this property.

[13]We need to determine the parameters of the variogram model: e.g. the nugget (C_0), sill (C_1) and the range (R) parameter. By knowing these parameters, we can estimate the semivariance at any location in the area of interest.

covariances at new location. Note that the $\mathbf{C}$ is in fact $(n+1) \times (n+1)$ matrix if it is used to derive kriging weights. One extra row and column are used to ensure that the sum of weights is equal to one:

$$
\begin{bmatrix}
C(\mathbf{s}_1, \mathbf{s}_1) & \cdots & C(\mathbf{s}_1, \mathbf{s}_n) & 1 \\
\vdots & & \vdots & \vdots \\
C(\mathbf{s}_n, \mathbf{s}_1) & \cdots & C(\mathbf{s}_n, \mathbf{s}_n) & 1 \\
1 & \cdots & 1 & 0
\end{bmatrix}^{-1}
\cdot
\begin{bmatrix}
C(\mathbf{s}_0, \mathbf{s}_1) \\
\vdots \\
C(\mathbf{s}_0, \mathbf{s}_n) \\
1
\end{bmatrix}
=
\begin{bmatrix}
w_1(\mathbf{s}_0) \\
\vdots \\
w_n(\mathbf{s}_0) \\
\varphi
\end{bmatrix}
\tag{1.3.5}
$$

where φ is the so-called *Langrange multiplier*.

In addition to estimation of values at new locations, a statistical spatial prediction technique offers a measure of associated uncertainty of making these estimations by using a given model. In geostatistics, this is often referred to as the **prediction variance**, i.e. the estimated variance of the prediction error. OK variance is defined as the weighted average of covariances from the new point ($\mathbf{s}_0$) to all calibration points ($\mathbf{s}_1, .., \mathbf{s}_n$), plus the Lagrange multiplier (Webster and Oliver, 2001, p.183):

$$
\begin{aligned}
\hat{\sigma}^2_{\mathsf{OK}}(\mathbf{s}_0) &= (C_0 + C_1) - \mathbf{c}_0^{\mathbf{T}} \cdot \lambda_0 \\
&= C_0 + C_1 - \sum_{i=1}^{n} w_i(\mathbf{s}_0) \cdot C(\mathbf{s}_0, \mathbf{s}_i) + \varphi
\end{aligned}
\tag{1.3.6}
$$

where $C(\mathbf{s}_0, \mathbf{s}_i)$ is the covariance between the new location and the sampled point pair, and φ is the Lagrange multiplier, as shown in Eq.(1.3.5).

As you can notice, outputs from any statistical prediction model are always two maps: (1) predictions and (2) prediction variance. The mean of the prediction variance at all location can be termed the **overall prediction variance**, and can be used as a measure of how precise is our final map: if the overall prediction variance gets close to the global variance, then the map is 100% imprecise; if the overall prediction variance tends to zero, then the map is 100% precise[14].

Note that a common practice in geostatistics is to model the variogram using a semivariance function and then, for the reasons of computational efficiency, use the **covariances**. In the case of solving the kriging weights, both the matrix of semivariances and covariances give the same results, so you should not really make a difference between the two. The relation between the covariances and semivariances is (Isaaks and Srivastava, 1989, p.289):

$$
C(\mathbf{h}) = C_0 + C_1 - \gamma(\mathbf{h})
\tag{1.3.7}
$$

where $C(\mathbf{h})$ is the covariance, and $\gamma(\mathbf{h})$ is the semivariance function (Fig. 1.8a). So for example, exponential model can be written in two ways:

$$
\gamma(\mathbf{h}) =
\begin{cases}
0 & \text{if} \quad |\mathbf{h}| = 0 \\
C_0 + C_1 \cdot \left[1 - e^{-\left(\frac{\mathbf{h}}{R}\right)}\right] & \text{if} \quad |\mathbf{h}| > 0
\end{cases}
\tag{1.3.8}
$$

$$
C(\mathbf{h}) =
\begin{cases}
C_0 + C_1 & \text{if} \quad |\mathbf{h}| = 0 \\
C_1 \cdot \left[e^{-\left(\frac{\mathbf{h}}{R}\right)}\right] & \text{if} \quad |\mathbf{h}| > 0
\end{cases}
\tag{1.3.9}
$$

---

[14]As we will see later on, the precision of mapping is measure of how well did we fit the point values. The true quality of map can only be accessed by using validation points, preferably independent from the point dataset used to make predictions.

The covariance at zero distance ($C(0)$) is by definition equal to the mean residual error (Cressie, 1993) — $C(\mathbf{h}_{11})$ also written as $C(\mathbf{s}_1, \mathbf{s}_1)$, and which is equal to $C(0) = C_0 + C_1 = Var\{z(s)\}$.
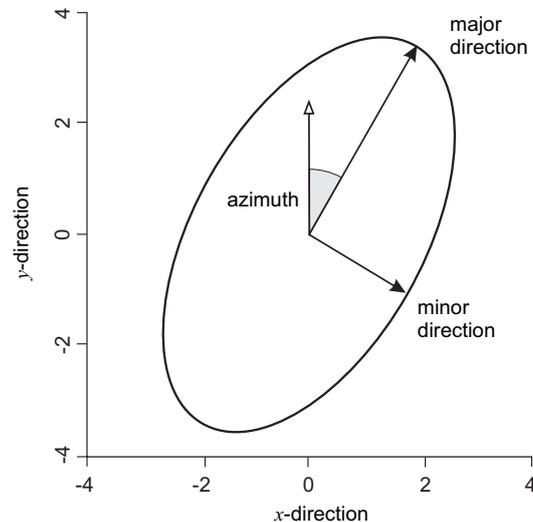

Fig. 1.9: Range ellipse for anisotropic model. After `gstat` User's manual.

The variogram models can be extended to even larger number of parameters if either (a) **anisotropy** or (b) smoothness are considered in addition to modelling of nugget and sill variation. The 2D geometric anisotropy in gstat, for example, is modelled by replacing the range parameter with three parameters — range in the major direction (direction of the strongest correlation), angle of the principal direction and the anisotropy ratio, e.g. (Fig. 1.9):

```
vgm(nugget=1, model="Sph", sill=10, range=2, anis=c(30,0.5))
```

where value of the angle of major direction is `30` (azimuthal direction measured in degrees clockwise), and value of the anisotropy ratio is `0.5` (range in minor direction is two times shorter).

Another sophistication of the standard 3-parameter variograms is the Matérn variogram model, which has an additional parameter to describe the smoothness (Stein, 1999; Minasny and McBratney, 2005):

$$\gamma(\mathbf{h}) = C_0 \cdot \delta(\mathbf{h}) + C_1 \cdot \left[\frac{1}{2^{v-1} \cdot \Gamma(v)} \cdot \left(\frac{\mathbf{h}}{R} \right)^v \cdot K_v \cdot \left(\frac{\mathbf{h}}{R} \right) \right] \tag{1.3.10}$$

where $\delta(\mathbf{h})$ is the Kronecker delta, K_v is the modified Bessel function, Γ is the gamma function and v is the smoothness parameter. The advantage of this model is that it can be used universally to model both short and long distance variation. In reality, variogram models with more parameters are more difficult to fit automatically because the iterative algorithms might get stuck in local minima (Minasny and McBratney, 2005). To avoid such problems, we will rely in §4 on more simple variogram models such as the Exponential model.

The fastest intuitive way to understand the principles of kriging is to use an educational program called **E*Z*-Kriging**, kindly provided by Dennis J.J. Walvoort from the Alterra Green World Research. The GUI of E*Z*-Kriging consists of three panels: (1)
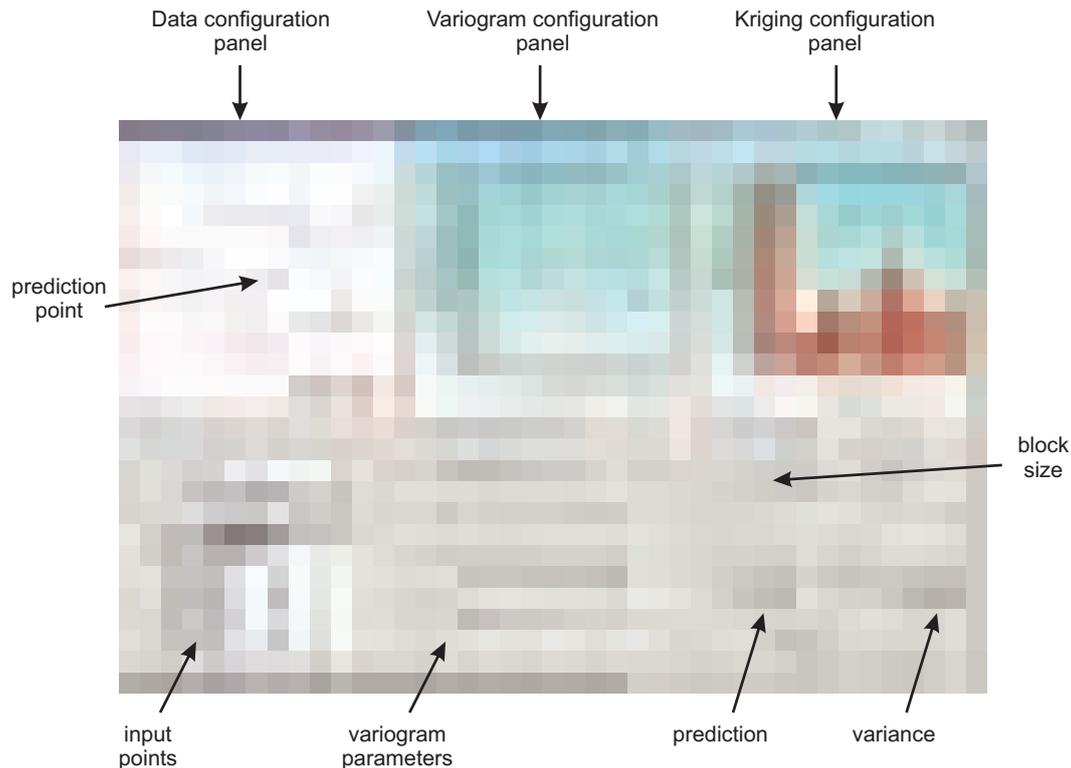
Fig. 1.10: Ordinary kriging explained: EZ-Kriging. Courtesy of Dennis J.J. Walvoort.

data configuration panel, (2) variogram panel, and (3) kriging panel (Fig. 1.10). This allows you to zoom into ordinary kriging and explore its main characterises and behaviour: how do weights change for different variogram models, how data values affect the weights, how does block size affects the kriging results etc. For example, if you study how model shape, nugget, sill and range affect the kriging results, you will notice that, assuming some standard variogram model (zero nugget, sill at global variance and practical range at 10% of the largest distance), the weights will decrease exponentially[15]. This is an important characteristic of kriging because it allows us to limit the search window to speed up the calculation and put more emphasise on fitting the semivariances at shorter distances. Note also that, although it commonly leads to smoothing of the values, kriging is an exact and non-convex interpolator. It is exact in the sense that the kriging estimates are equal to input values at sampling locations, and it is non-convex because its predictions can be outside the data range, e.g. we can produce negative concentrations.

Another important aspect of using kriging is the issue of the support size. In geostatistics, one can control the support size of the outputs by averaging multiple (randomized) point predictions over regular blocks of land. This is known as **block prediction** (Heuvelink and Pebesma, 1999). A problem is that we can sample elevation at point locations, and then interpolate them for blocks of e.g. 10×10 m, but we could also take composite samples and interpolate them at point locations. This often confuses GIS users because as well as using point measurements to interpolate values at regular point

---

[15]In practice, often >95% of weights will be explained by the nearest 30–50 points. Only if the variogram is close to the pure nugget model, the more distant points will receive more importance, but then the technique will produce poor predictions anyhow.

locations (e.g. by point kriging), and then display them using a raster map (see Fig. 1.6), we can also make spatial predictions for blocks of land (block kriging) and display them using the same raster model (Bishop and McBratney, 2001). For simplicity, in the case of block-kriging, one should always use the cell size that corresponds to the support size.

1.3.2 Environmental correlation

If some exhaustively-sampled auxiliary variables or **covariates** are available in the area of interest and if they are significantly correlated with our target variable (spatial cross-correlation), and assuming that the point-values are not spatially auto-correlated, predictions can be obtained by focusing only on the deterministic part of variation:

$$Z(\mathbf{s}) = f\{q_k(\mathbf{s})\} + \varepsilon \tag{1.3.11}$$

where q_k are the auxiliary predictors that can be used to explain the deterministic part of spatial variation. This approach to spatial prediction has a strong physical interpretation. Consider Rowe and Barnes (1994) observation that earth surface energy-moisture regimes at all scales/sizes are the dynamic driving variables of functional ecosystems at all scales/sizes. The concept of vegetation/soil-environment relationships has frequently been presented in terms of an equation with six key **environmental factors** as:

$$V \times S[x, y, \tilde{t}] = f\left\{ \begin{array}{lll} s[x, y, \tilde{t}] & c[x, y, \tilde{t}] & o[x, y, \tilde{t}] \\ r[x, y, \tilde{t}] & p[x, y, \tilde{t}] & a[x, y, \tilde{t}] \end{array} \right. \tag{1.3.12}$$

where V stands for vegetation, S for soil, c stands for climate, o for organisms (including humans), r is relief, p is parent material or geology, a is age of the system, x, y are the coordinates and t is time dimension. This means that the predictors which are available over entire areas of interest can be used to predict the value of an environmental variable at unvisited locations — first by modelling the relationship between the target and auxiliary environmental predictors at sample locations, and then by applying it to unvisited locations using the known value of the auxiliary variables at those locations. Common auxiliary environmental predictors used to map environmental variables are land surface parameters, remote sensing images, and geological, soil and land-use maps (McKenzie and Ryan, 1999). Because many auxiliary predictors (see further Table 3.2) are now also available at low or no cost, it makes this approach to spatial prediction ever more important (Hengl et al., 2007b).

Functional relations between environmental variables and factors are in general unknown and the correlation coefficients can differ for different study areas, different seasons and different scales. However, in many cases, relations with the environmental predictors often reflect causal linkage: deeper and more developed soils occur at places of higher potential accumulation and lower slope; different type of forests can be found at different expositions and elevations; soils with more organic matter can be found where the climate is cooler and wetter etc. This makes this technique especially suitable for natural resource inventory teams because it allows them to validate their empirical knowledge about the variation of the target features in the area of interest.

There are (at least) four groups of statistical models that have been used to make spatial predictions with the help of environmental factors (Chambers and Hastie, 1992; McBratney et al., 2003; Bishop and Minasny, 2005):

Classification-based models — Classification models are primarily developed and used when we are dealing with discrete target variables (e.g. land cover or soil

types). There is also a difference whether **Boolean** (crisp) or **Fuzzy** (continuous) classification rules are used to create outputs. Outputs from the model fitting process are class boundaries (class centres and standard deviations) or classification rules.

Tree-based models — Tree-based models are often easier to interpret when a mix of continuous and discrete variables are used as predictors (Chambers and Hastie, 1992). They are fitted by successively splitting a dataset into increasingly homogeneous groupings. Output from the model fitting process is a **decision tree**, which can then be applied to make predictions of either individual property values or class types for an entire area of interest.

Regression models — Regression analysis employs a family of functions called **Generalized Linear Models** (GLMs), which all assume a linear relationship between the inputs and outputs (Neter et al., 1996). Output from the model fitting process is a set of regression coefficients. Regression models can be also used to represent non-linear relationships with the use of **General Additive Models** (GAMs). The relationship between the predictors and targets can be solved using one-step data-fitting or by using iterative data fitting techniques (neural networks and similar).

Each of the models listed above can be equally applicable for mapping of environmental variables and can exhibit advantages and disadvantages. For example, some advantages of using tree-based regression are that they: can handle missing values, can use continuous and categorical predictors, are robust to predictor specification, and make very limited assumptions about the form of the regression model (Henderson et al., 2004). Some disadvantages of regression trees, on the other hand, is that they require large datasets and completely ignore spatial position of the input points.

A common regression-based approach to spatial prediction is the **multiple linear regression** (Draper and Smith, 1998). Here, the predictions are again obtained by weighted averaging (compare with Eq.(1.3.2)), this time by averaging the predictors:

$$\hat{z}_{\texttt{OLS}}(\mathbf{s}_0) = \hat{b}_0 + \hat{b}_1 \cdot q_1(\mathbf{s}_0) + \ldots + \hat{b}_p \cdot q_p(\mathbf{s}_0) = \sum_{k=0}^{p} \hat{\beta}_k \cdot q_k(\mathbf{s}_0); \qquad q_0(\mathbf{s}_0) \equiv 1 \quad (1.3.13)$$

or in matrix algebra:

$$\hat{z}_{\texttt{OLS}}(\mathbf{s}_0) = \beta^{\mathbf{T}} \cdot \mathbf{q} \qquad (1.3.14)$$

where $q_k(\mathbf{s}_0)$ are the values of the auxiliary variables at the target location, p is the number of predictors or auxiliary variables[16], and $\hat{\beta}_k$ are the regression coefficients solved using the **Ordinary Least Squares**:

$$\hat{\beta} = \left(\mathbf{q}^{\mathbf{T}} \cdot \mathbf{q}\right)^{-\mathbf{1}} \cdot \mathbf{q}^{\mathbf{T}} \cdot \mathbf{z} \qquad (1.3.15)$$

where $\mathbf{q}$ is the matrix of predictors ($n \times p+1$) and $\mathbf{z}$ is the vector of sampled observations. The prediction error of a multiple linear regression model is (Neter et al., 1996, p.210):

$$\hat{\sigma}^2_{\texttt{OLS}}(\mathbf{s}_0) = MSE \cdot \left[1 + \mathbf{q}_0^{\mathbf{T}} \cdot \left(\mathbf{q}^{\mathbf{T}} \cdot \mathbf{q}\right)^{-\mathbf{1}} \cdot \mathbf{q}_0\right] \qquad (1.3.16)$$

---

[16]To avoid confusion with geographical coordinates, we use the symbol q, instead of the more common x, to denote a predictor.
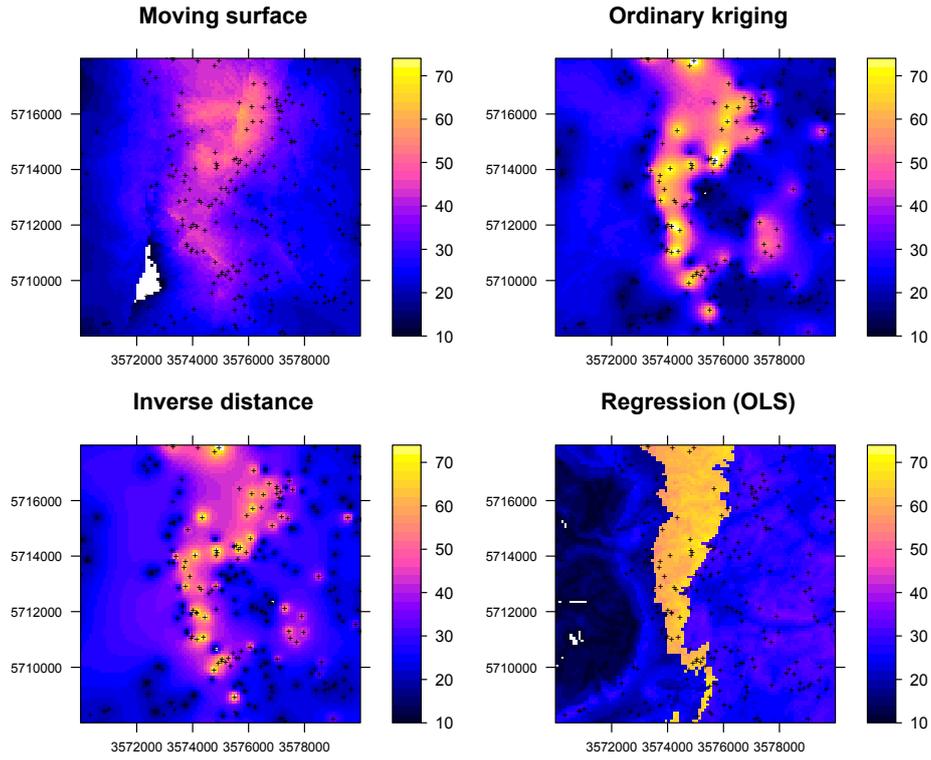
Fig. 1.11: Comparison of spatial prediction techniques for mapping SAND (%) in topsoil (see further §4.7). Note that inverse distance interpolation and kriging are often quite similar, while the moving trend surface (2nd order polynomial) can lead to artefacts (negative values) — locally where the density of points is poor. The regression-based (OLS) predictions were produced using a DEM, wetness index and geological mapping units as predictors.

where MSE is the mean square (residual) error around the regression line:

$$MSE = \frac{\sum\limits_{i=1}^{n} \left[z(\mathbf{s}_i) - \hat{z}(\mathbf{s}_i) \right]^2}{n-2} \tag{1.3.17}$$

and $\mathbf{q}_0$ is the vector of predictors at new, unvisited location. In the univariate case, the variance of the prediction error can also be derived using:

$$\hat{\sigma}^2(\mathbf{s}_0) = MSE \cdot \left[1 + \frac{1}{n} + \frac{\left[q(\mathbf{s}_0) - \bar{q} \right]^2}{\sum\limits_{i=1}^{n} \left[q(\mathbf{s}_i) - \bar{q} \right]^2} \right] = MSE \cdot \left[1 + v(\mathbf{s}_0) \right] \tag{1.3.18}$$

where v is the curvature of the confidence band around the regression line. This reflects the amount of extrapolation in the feature space (Ott and Longnecker, 2001, p.570). It can be seen from Eq. (1.3.18) that the prediction error, for a given n (sampling intensity), depends on three factors:

(1.) Mean square residual error (MSE);

(2.) Spreading of points in the feature space $\sum \left[q(\mathbf{s}_i) - \bar{q} \right]^2$;

(3.) 'Distance' of the new observation from the centre of the feature space $\left[q(\mathbf{s}_0) - \bar{q} \right]$.

So in general, if the model is linear, we can decrease the prediction variance if we increase the spreading of the points in features space. Understanding this principles allows us to prepare sampling plans that will achieve higher mapping precision and minimize extrapolation in feature space (see further §2.6).

The sum of squares of residuals (SSE) can be used to determine the **adjusted coefficient of multiple determination** (R_a^2), which describes the goodness of fit:

$$
\begin{aligned}
R_a^2 &= 1 - \left(\frac{n-1}{n-p}\right) \cdot \frac{SSE}{SSTO} \\
&= 1 - \left(\frac{n-1}{n-p}\right) \cdot \left(1 - R^2\right)
\end{aligned}
\tag{1.3.19}
$$

where $SSTO$ is the total sum of squares (Neter et al., 1996), R^2 indicates amount of variance explained by model, whereas R_a^2 adjusts for the number of variables (p) used. For many environmental mapping projects, a $R_a^2 \geq 0.85$ is already a very satisfactory solution and higher values will typically only mean over-fitting of the data (Park and Vlek, 2002).

The principle of predicting environmental variables using factors of climate, relief, geology and similar, is often referred to as **environmental correlation**. The *environmental correlation approach* to mapping is a true alternative to ordinary kriging (compare the produced patterns in Fig. 1.11). This is because both approaches deal with different aspects of spatial variation: regression deals with the deterministic and kriging with the spatially-correlated stochastic part of variation.

The biggest criticism of pure regression approach to spatial prediction is that the position of points in the geographical space is completely ignored, both during the model fitting and prediction. Imagine if we are dealing with two point datasets where one data set is heavily clustered, while the other is well-spread over the area of interest — these has to be a way to account for the clustering of the points so we take the model derived using the clustered points with much bigger caution.

One way to account for this problem is to take the distance between the points into account during the estimation of the regression coefficients. This can be achieved by using the **geographically weighted regression** (Fotheringham et al., 2002). So instead of using the OLS estimation (Eq.1.3.15) we use:

$$
\hat{\beta}_{\texttt{WLS}} = \left(\mathbf{q^T} \cdot \mathbf{W} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q^T} \cdot \mathbf{W} \cdot \mathbf{z}
\tag{1.3.20}
$$

where $\mathbf{W}$ is a matrix of weights, determined using some distance decay function e.g.:

$$
w_i(\mathbf{s}_i, \mathbf{s}_j) = \sigma_E^2 \cdot \exp\left[-3 \cdot \frac{d^2(\mathbf{s}_i, \mathbf{s}_j)}{\daleth^2}\right]
\tag{1.3.21}
$$

where σ_E^2 is the level of variation of the error terms, $d(\mathbf{s}_i, \mathbf{s}_j)$ is the Euclidian distance between a sampled point pair and $\daleth$ is known as the bandwidth, which determines the degree of *locality* — small values of $\daleth$ suggest that correlation only occurs between very close point pairs and large values suggest that such effects exist even on a larger spatial scale. Compare further with Eq.(2.1.3).

1.3.3 Predicting from polygon maps

A special case of environmental correlation is prediction from polygon maps i.e. stratified areas (different land use/cover types, geological units etc). Assuming that the residuals

show no spatial auto-correlation, a value at new location can be predicted by:

$$\hat{z}(\mathbf{s}_0) = \sum_{i=1}^{n} w_i \cdot z(s); \qquad w_i = \begin{cases} 1/n_k & \text{for } x_i \in k \\ 0 & \text{otherwise} \end{cases} \qquad (1.3.22)$$

where k is the unit identifier. This means that the weights within some unit will be equal so that the predictions are made by simple averaging per unit (Webster and Oliver, 2001):

$$\hat{z}(\mathbf{s}_0) = \bar{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} z(\mathbf{s}_i) \qquad (1.3.23)$$

Consequently, the output map will show only abrupt changes in the values between the units. The prediction variance of this prediction model is simply the within-unit variance:

$$\hat{\sigma}^2(\mathbf{s}_0) = \frac{\sigma_k^2}{n_k} \qquad (1.3.24)$$

From Eq.(1.3.24) it is obvious that the precision of the technique will be maximized if the within-unit variation is infinitely small. Likewise, if the within-unit variation is as high as the global variability, the predictions will be as bad as predicting by taking any value from the normal distribution.

Another approach to make predictions from polygon maps is to use multiple regression. In this case, the predictors (mapping units) are used as indicators:

$$\hat{z}(\mathbf{s}_0) = \hat{b}_1 \cdot MU_1(\mathbf{s}_0) + \ldots + \hat{b}_k \cdot MU_k(\mathbf{s}_0); \qquad MU_k \in [0|1] \qquad (1.3.25)$$

and it can be shown that the OLS fitted regression coefficients will equal the mean values within each strata ($b_k = \bar{\mu}(MU_k)$), so that the Eqs.(1.3.25) and (1.3.23) are in fact equivalent.

If, on the other hand, the residuals do show spatial auto-correlation, the predictions can be obtained by **stratified kriging**. This is basically ordinary kriging done separately for each strata and can often be impractical because we need to estimate a variogram for each of the k strata (Boucneau et al., 1998). Note that the strata or sub-areas need to be known *a priory* and they should never be derived from the data used to generate spatial predictions.

1.3.4 Mixed or hybrid models

Mixed or hybrid spatial prediction models comprise of a combination of the techniques listed previously. For example, a mixed geostatistical model employs both correlation with auxiliary predictors and spatial autocorrelation simultaneously. There are two main sub-groups of mixed geostatistical models: (a) **co-kriging**-based and (b) **regression-kriging**-based techniques (Goovaerts, 1997), but the list could certainly be extended.

Note also that, in the case of environmental correlation by linear regression, we assume some basic (additive) model, although the relationship can be much more complex. To account for this, a linear regression model can be extended to a diversity of statistical models ranging from regression trees, General Additive Models, neural networks and similar. Consequently, the mixed models are more generic than pure kriging-based or regression-based techniques and can be used to represent both discrete and continuous changes in the space, both deterministic and stochastic processes.

One can also combine deterministic, statistical and expert-based estimation models. For example, one can use a deterministic model to estimate a value of the variable, then use actual measurements to fit a calibration model, analyse the residuals for spatial correlation and eventually combine the statistical fitting and deterministic modelling (Hengl et al., 2007b). Most often, expert-based models are supplemented with the actual measurements, which are then used to refine the rules, e.g. using the neural networks (Kanevski et al., 1997).

Important sources:

★ Rossiter D.G., 2005. Geostatistics, lecture notes, ITC, Enschede, Netherlands.

★ Nielsen, D. and Wendroth, O., 2003. Spatial and Temporal Statistics — Sampling Field Soils and Their Vegetation. Catena-Verlag, Reiskirchen, 614 pp.

★ Webster, R. and Oliver, M.A., 2001. Geostatistics for Environmental Scientists. Statistics in Practice. John Wiley & Sons, Chichester, 265 pp.

★ Goovaerts, P., 1997. Geostatistics for Natural Resources Evaluation (Applied Geostatistics). Oxford University Press, New York, 496 pp.

★ Isaaks, E.H. and Srivastava, R.M. 1989. An Introduction to Applied Geostatistics. Oxford University Press, New York, 542 pp.

★ http://www.wiley.co.uk/eoenv/ — The Encyclopedia of Environmetrics.

★ http://geoenvia.org — A research association that promotes use of geostatistical methods for environmental applications.

2

Regression-kriging

As we saw in the previous chapter, there seems to be many possibilities to map environmental variables using geostatistics. In reality, we always try to go for the most flexible, most comprehensive and the most robust technique (preferably implemented in a software with an user-friendly GUI). In fact, many (geo)statisticians believe that there is only one Best Linear Unbiased Prediction (**BLUP**) model for spatial data (Gotway and Stroup, 1997; Stein, 1999; Christensen, 2001). As we will see further on in this chapter, one such generic mapping technique is the regression-kriging. All other techniques mentioned previously — ordinary kriging, environmental correlation, averaging of values per polygons or inverse distance interpolation — can be seen as its special cases.

2.1 The Best Linear Unbiased Predictor of spatial data

Matheron (1969) proposed that a value of a target variable at some location can be modelled as a sum of the deterministic and stochastic components:

$$Z(\mathbf{s}) = m(\mathbf{s}) + \varepsilon'(\mathbf{s}) + \varepsilon'' \qquad (2.1.1)$$
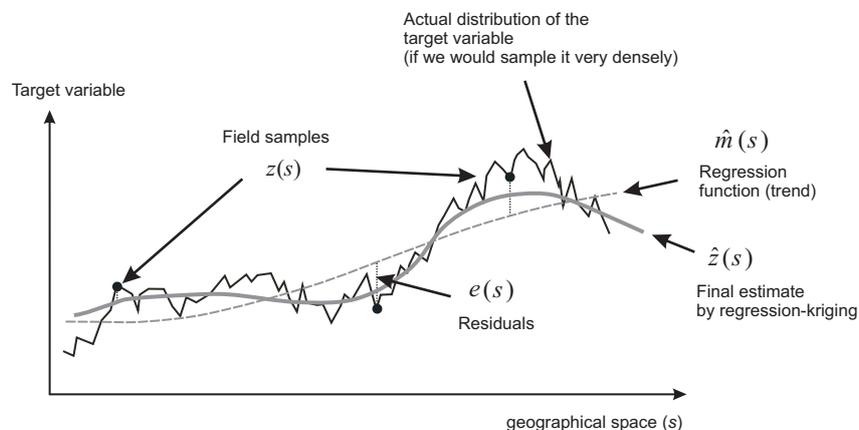
Fig. 2.1: A schematic example of regression-kriging: fitting a vertical cross-section with assumed distribution of an environmental variable in horizontal space.

which he termed **universal model of spatial variation**. We have seen in the previous sections (§1.3.1 and §1.3.2) that both deterministic and stochastic components of spatial variation can be modelled separately. By combining the two approaches, we obtain:

$$
\begin{aligned}
\hat{z}(\mathbf{s}_0) &= \hat{m}(\mathbf{s}_0) + \hat{e}(\mathbf{s}_0) \\
&= \sum_{k=0}^{p} \hat{\beta}_k \cdot q_k(\mathbf{s}_0) + \sum_{i=1}^{n} \lambda_i \cdot e(\mathbf{s}_i)
\end{aligned}
\tag{2.1.2}
$$

where $\hat{m}(\mathbf{s}_0)$ is the fitted deterministic part, $\hat{e}(\mathbf{s}_0)$ is the interpolated residual, $\hat{\beta}_k$ are estimated deterministic model coefficients ($\hat{\beta}_0$ is the estimated intercept), λ_i are kriging weights determined by the spatial dependence structure of the residual and where $e(\mathbf{s}_i)$ is the residual at location $\mathbf{s}_i$. The regression coefficients $\hat{\beta}_k$ can be estimated from the sample by some fitting method, e.g. ordinary least squares (OLS) or, optimally, using **Generalized Least Squares** (Cressie, 1993, p.166):

$$
\hat{\beta}_{\text{GLS}} = \left(\mathbf{q}^{\mathbf{T}} \cdot \mathbf{C}^{-1} \cdot \mathbf{q} \right)^{-1} \cdot \mathbf{q}^{\mathbf{T}} \cdot \mathbf{C}^{-1} \cdot \mathbf{z}
\tag{2.1.3}
$$

where $\hat{\beta}_{\text{GLS}}$ is the vector of estimated regression coefficients, $\mathbf{C}$ is the covariance matrix of the residuals, $\mathbf{q}$ is a matrix of predictors at the sampling locations and $\mathbf{z}$ is the vector of measured values of the target variable. The GLS estimation of regression coefficients is, in fact, a special case of the geographically weighted regression (compare with Eq.1.3.20). In the case, the weights are determined objectively to account for the spatial auto-correlation between the residuals.

Once the deterministic part of variation has been estimated, the residual can be interpolated with kriging and added to the estimated trend (Fig. 2.1). The estimation of the residuals is an iterative process: first the deterministic part of variation is estimated using ordinary least squares (OLS), then the covariance function of the residuals is used to obtain the GLS coefficients. Next, these are used to re-compute the residuals, from which an updated covariance function is computed, and so on. Although this is by many geostatisticians recommended as the proper procedure, Kitanidis (1994) showed that use of the covariance function derived from the OLS residuals (i.e. a single iteration) is often satisfactory, because it is not different enough from the function derived after several iterations; i.e. it does not affect much the final predictions. Minasny and McBratney (2007) recently reported similar results — it is much more important to use more useful and higher quality data then to use more sophisticated statistical methods.

In matrix notation, regression-kriging is commonly written as (Christensen, 2001, p.277):

$$
\hat{z}_{\text{RK}}(\mathbf{s}_0) = \mathbf{q_0^T} \cdot \hat{\beta}_{\text{GLS}} + \lambda_0^{\mathbf{T}} \cdot (\mathbf{z} - \mathbf{q} \cdot \hat{\beta}_{\text{GLS}})
\tag{2.1.4}
$$

where $\hat{z}(\mathbf{s}_0)$ is the predicted value at location $\mathbf{s}_0$, $\mathbf{q_0}$ is the vector of $p + 1$ predictors and λ_0 is the vector of n kriging weights used to interpolate the residuals. The model in Eq.(2.1.4) is considered to be the Best Linear Predictor of spatial data. It has a prediction variance that reflects the position of new locations (extrapolation) in both geographical and feature space:

$$
\begin{aligned}
\hat{\sigma}_{\text{RK}}^2(\mathbf{s}_0) = {}&(C_0 + C_1) - \mathbf{c_0^T} \cdot \mathbf{C}^{-1} \cdot \mathbf{c_0} \\
&+ \left(\mathbf{q_0} - \mathbf{q^T} \cdot \mathbf{C}^{-1} \cdot \mathbf{c_0} \right)^{\mathbf{T}} \cdot \left(\mathbf{q^T} \cdot \mathbf{C}^{-1} \cdot \mathbf{q} \right)^{-1} \cdot \left(\mathbf{q_0} - \mathbf{q^T} \cdot \mathbf{C}^{-1} \cdot \mathbf{c_0} \right)
\end{aligned}
\tag{2.1.5}
$$
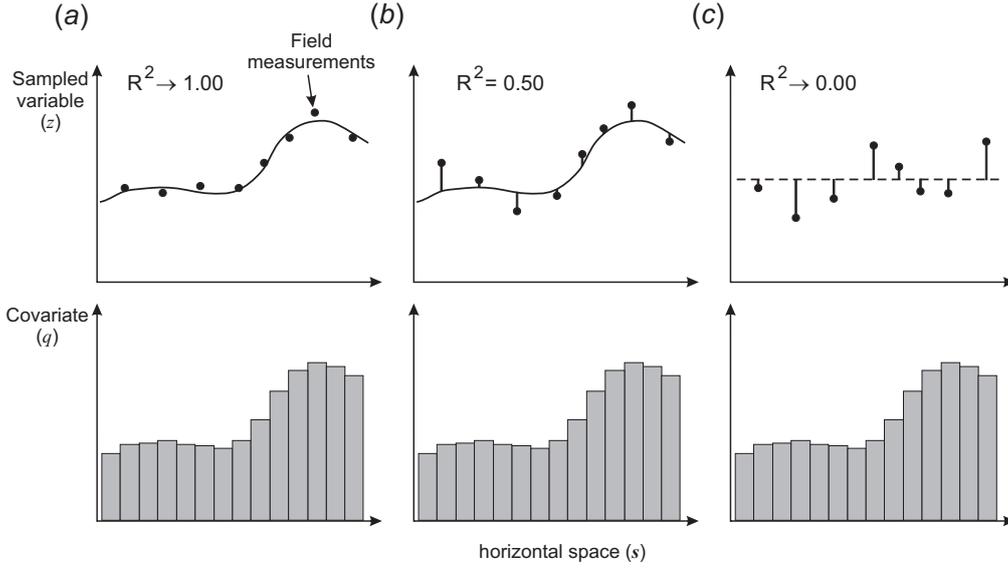
Fig. 2.2: Whether we will use pure regression model, pure kriging or hybrid regression-kriging is basically determined by R-square: (a) if R-square is very high, then the residuals will be infinitively small; (c) if R-square is insignificant, then we will probably finish with using ordinary kriging; (b) in most cases, we will use a combination of regression and kriging.

where $C_0 + C_1$ is the sill variation and $\mathbf{c_0}$ is the vector of covariances of residuals at the unvisited location.

Obviously, if the residuals show no spatial auto-correlation (pure nugget effect), the regression-kriging (Eq.2.1.4) converges to pure multiple linear regression (Eq.1.3.14) because the covariance matrix ($\mathbf{C}$) becomes identity matrix:

$$\mathbf{C} = \begin{bmatrix} C_0 + C_1 & \cdots & 0 \\ \vdots & C_0 + C_1 & 0 \\ 0 & 0 & C_0 + C_1 \end{bmatrix} = (C_0 + C_1) \cdot \mathbf{I} \qquad (2.1.6)$$

so the kriging weights (Eq.1.3.4) at any location predict the mean residual i.e. 0 value. Similarly, the regression-kriging variance (Eq.2.1.5) reduces to the multiple linear regression variance (Eq.1.3.16):

$$\sigma_{\text{RK}}^2(\mathbf{s}_0) = (C_0 + C_1) - 0 + \mathbf{q_0^T} \cdot \left(\mathbf{q^T} \cdot \frac{1}{(C_0 + C_1)} \cdot \mathbf{q} \right)^{-1} \cdot \mathbf{q_0}$$

$$\sigma_{\text{RK}}^2(\mathbf{s}_0) = (C_0 + C_1) + (C_0 + C_1) \cdot \mathbf{q_0^T} \cdot \left(\mathbf{q^T} \cdot \mathbf{q} \right)^{-1} \cdot \mathbf{q_0}$$

and since $(C_0 + C_1) = C(0) = MSE$, the RK variance reduces to the MLR variance:

$$\hat{\sigma}_{\text{RK}}^2(\mathbf{s}_0) = \hat{\sigma}_{\text{OLS}}^2(\mathbf{s}_0) = MSE \cdot \left[1 + \mathbf{q_0^T} \cdot \left(\mathbf{q^T} \cdot \mathbf{q} \right)^{-1} \cdot \mathbf{q_0} \right] \qquad (2.1.7)$$

Likewise, if the target variable shows no correlation with the auxiliary predictors, the regression-kriging model reduces to ordinary kriging model because the deterministic part equals the (global) mean value (Fig. 2.2c, Eq.1.3.25).

The formulas above show that, depending on the strength of the correlation, the RK might turn to pure kriging — if predictors are uncorrelated with the target variable —

or pure regression — if there is significant correlation and the residuals show pure nugget variogram (Fig. 2.2). Hence, pure kriging and pure regression should be considered as only special cases of regression-kriging (Hengl et al., 2004a, 2007b).

2.1.1   Selecting the right spatial prediction technique

Knowing that the most of the linear spatial prediction models are more or less connected, we can start from testing the most generic technique, and then finish with using the most suitable technique. Pebesma (2004, p.689), for example, implemented such nested structure in his design of the `gstat` package. An user can switch between one to other technique by following a simple decision tree (Fig. 2.3). First, we need to check if the deterministic model is defined already, if it has not been, we can try to correlate the sampled variables with environmental factors. If the environmental factors are significantly correlated, we can fit a multiple linear regression model (Eq.1.3.14) and then analyse the residuals for spatial autocorrelation. If the residuals show no spatial autocorrelation (pure nugget effect), we proceed with OLS estimation of the regression coefficients. Otherwise, if the residuals show spatial auto-correlation, we can run regression-kriging. If the data shows no correlation with environmental factors, then we can still analyse the variogram of the target variable. This time, we might also consider modelling the anisotropy. If we can fit a variogram different from pure nugget effect, then we can run ordinary kriging. Otherwise, if we can only fit a linear variogram, then we might just use the inverse distance interpolation.

If the variogram of the target variable shows no spatial auto-correlation, and no correlation with environmental factors, this practically means that the only statistically valid prediction model is to estimate a global mean for the whole area. Although this might frustrate you because it would lead to a non-sense map where each pixel shows the same value, you should be aware that even this is informative[1].

How does the selection of the spatial prediction model works in practice? In the `gstat` package, a user can easily switch from one to other prediction model by changing the arguments in the generic `krige` function in R (see further §3.1.3). For example, if the name of the input field samples is `points` and the grid is defined by `mapgrid`, we can run the inverse distance interpolation (§1.2.1) by specifying (Pebesma, 2004):

```
ev.id = krige(ev~1, data=points, newdata=mapgrid)
```

where `ev` is the sampled environmental variable (vector) and `ev.id` is the resulting raster map (see Fig. 1.11 for an example). Instead of using inverse distance interpolation we might also try to fit the values using the coordinates and a 2nd order polynomial model:

```
ev.ts = krige(ev~x+y+x*y+x*x+y*y, data=points, newdata=mapgrid)
```

which can be converted to the moving surface fitting by adding a search window (Fig. 1.11):

```
ev.mv = krige(ev~x+y+x*y+x*x+y*y, data=points, newdata=mapgrid, nmax=20)
```

If we add a variogram model, then `gstat` will instead of running inverse distance interpolation run ordinary kriging (§1.3.1):

```
ev.ok = krige(ev~1, data=points, newdata=mapgrid, model=vgm(psill=5,
"Exp", range=1000, nugget=1))
```

---

[1]Sometimes an information that we are completely uncertain about a feature is better than a colorful but completely unreliable map.
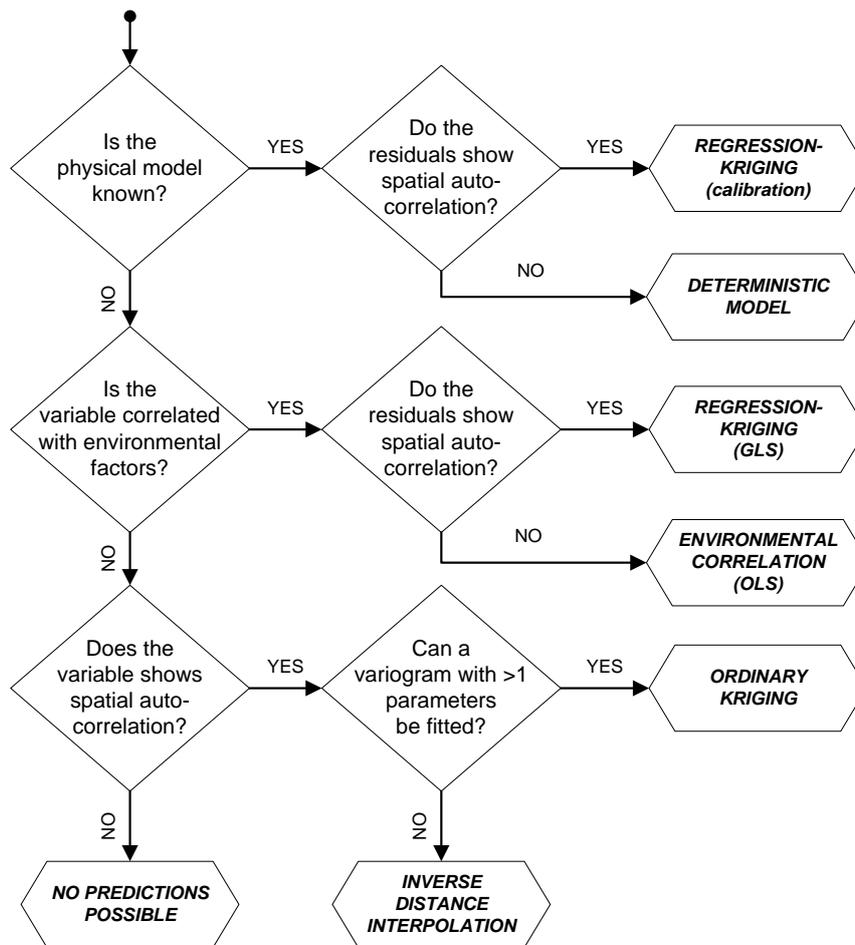
Fig. 2.3: Decision tree for selecting a suitable spatial prediction model.

where `vgm(psill=5, "Exp", range=1000, nugget=1)` is the Exponential variogram model with a sill parameter of 5, range parameter of 1000 m and the nugget parameter of 1. Likewise, if there were environmental factors significantly correlated with the target variable, we could run OLS regression (§1.3.2) by omitting the variogram model:

```
ev.ec = krige(ev~q1+q2, data=points, newdata=mapgrid)
```

where `q1` and `q2` are environmental factors used as predictors (raster maps), which are available as separate layers within the spatial layer[2] `mapgrid`. If the residuals do show spatial auto-correlation, then we can switch to regression-kriging (§2.1) by adding the variogram:

```
ev.rk = krige(ev~q1+q2, data=points, newdata=mapgrid, model=vgm(psill=3,
"Exp", range=500, nugget=0))
```

If the model between the environmental factors and our target variable is deterministic, then we can use the point samples to calibrate our predictions (assuming that the residuals show spatial auto-correlation). The R command would then look something like this:

---

[2]A grid data layer with multiple bands in R is called `SpatialGridDataframe`.

```
ev.rkc = krige(ev~ev.df, data=points, newdata=mapgrid,
model=vgm(psill=3, "Exp", range=500, nugget=0))
```

where `ev.df` are the values of the target variable estimated using a deterministic function.

In `gstat`, a user can also easily switch from estimation to simulations (§2.4) by adding to the command above an additional argument: `nsim=1`. This will generate Sequential Gaussian Simulations using the same prediction model. Multiple simulations can be generated by increasing the number set for this argument. In addition, a user can switch from block predictions by adding an argument, e.g. `block=100`; and from global estimation of weights by adding a search radius or maximum number of pairs, e.g. `radius=1000` or `nmax=60`.

2.1.2   Universal kriging, kriging with external drift

The geostatistical literature uses many different terms for what are essentially the same or at least very similar techniques. This confuses the users and distracts them from using the right technique for their mapping projects. In this section, we will show that both universal kriging, kriging with external drift and regregression-kriging are basically the same technique. Matheron (1969) originally termed the technique *Le krigeage universel*, however, the technique was intended as a generalized case of kriging where the trend is modelled as a function of coordinates. Thus, many authors (Deutsch and Journel, 1998; Wackernagel, 2003; Papritz and Stein, 1999) reserve the term *Universal Kriging* (UK) for the case when only the coordinates are used as predictors. If the deterministic part of variation (*drift*) is defined externally as a linear function of some auxiliary variables, rather than the coordinates, the term *Kriging with External Drift* (KED) is preferred (Wackernagel, 2003; Chiles and Delfiner, 1999). In the case of UK or KED, the predictions are made as with kriging, with the difference that the covariance matrix of residuals is extended with the auxiliary predictors $q_k(\mathbf{s}_i)$'s (Webster and Oliver, 2001, p.183). However, the drift and residuals can also be estimated separately and then summed. This procedure was suggested by Ahmed and de Marsily (1987) and Odeh et al. (1995) later named it *Regression-kriging*, while Goovaerts (1997, §5.4) uses the term *Kriging with a trend model* to refer to a family of interpolator, and refers to RK as *Simple kriging with varying local means*. Although equivalent, KED and RK differ in the computational steps used.

Let us zoom into the two variants of regression-kriging. In the case of KED, predictions at new locations are made by:

$$\hat{z}_{\text{KED}}(\mathbf{s}_0) = \sum_{i=1}^{n} w_i^{\text{KED}}(\mathbf{s}_0) \cdot z(\mathbf{s}_i) \qquad (2.1.8)$$

for

$$\sum_{i=1}^{n} w_i^{\text{KED}}(\mathbf{s}_0) \cdot q_k(\mathbf{s}_i) = q_k(\mathbf{s}_0); \qquad k = 1, ..., p \qquad (2.1.9)$$

or in matrix notation:

$$\hat{z}_{\text{KED}}(\mathbf{s}_0) = \delta_{\mathbf{0}}^{\mathbf{T}} \cdot \mathbf{z} \qquad (2.1.10)$$

where z is the target variable, q_k's are the predictor variables i.e. values at a new location ($\mathbf{s}_0$), $\delta_{\mathbf{0}}$ is the vector of KED weights (w_i^{KED}), p is the number of predictors and $\mathbf{z}$ is the

vector of n observations at primary locations. The KED weights are solved using the extended matrices:

$$\lambda_0^{\text{KED}} = \left\{ w_1^{\text{KED}}(\mathbf{s}_0), ..., w_n^{\text{KED}}(\mathbf{s}_0), \varphi_0(\mathbf{s}_0), ..., \varphi_p(\mathbf{s}_0) \right\}^{\mathbf{T}}$$

$$= \mathbf{C}^{\text{KED}-1} \cdot \mathbf{c}_0^{\text{KED}} \qquad (2.1.11)$$

where λ_0^{KED} is the vector of solved weights, φ_p are the Lagrange multipliers, $\mathbf{C}^{\text{KED}}$ is the extended covariance matrix of residuals and $\mathbf{c}_0^{\text{KED}}$ is the extended vector of covariances at new location.

In the case of KED, the extended covariance matrix of residuals looks like this (Webster and Oliver, 2001, p.183):

$$\mathbf{C}^{\text{KED}} = \begin{bmatrix} C(\mathbf{s}_1, \mathbf{s}_1) & \cdots & C(\mathbf{s}_1, \mathbf{s}_n) & 1 & q_1(\mathbf{s}_1) & \cdots & q_p(\mathbf{s}_1) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ C(\mathbf{s}_n, \mathbf{s}_1) & \cdots & C(\mathbf{s}_n, \mathbf{s}_n) & 1 & q_1(\mathbf{s}_n) & \cdots & q_p(\mathbf{s}_n) \\ 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ q_1(\mathbf{s}_1) & \cdots & q_1(\mathbf{s}_n) & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & 0 & \vdots & & \vdots \\ q_p(\mathbf{s}_1) & \cdots & q_p(\mathbf{s}_n) & 0 & 0 & \cdots & 0 \end{bmatrix} \qquad (2.1.12)$$

and $\mathbf{c}_0^{\text{KED}}$ like this:

$$\mathbf{c}_0^{\text{KED}} = \left\{ C(\mathbf{s}_0, \mathbf{s}_1), ..., C(\mathbf{s}_0, \mathbf{s}_n), q_0(\mathbf{s}_0), q_1(\mathbf{s}_0), ..., q_p(\mathbf{s}_0) \right\}^{\mathbf{T}}; \quad q_0(\mathbf{s}_0) = 1 \qquad (2.1.13)$$

Hence, KED looks exactly as ordinary kriging (Eq.1.3.2), except the covariance matrix/vector are extended with values of auxiliary predictors.

In the case of RK, the predictions are made separately for the drift and residuals and then added back together:

$$\hat{z}_{\text{RK}}(\mathbf{s}_0) = \mathbf{q}_0^{\mathbf{T}} \cdot \hat{\beta}_{\text{GLS}} + \lambda_0^{\mathbf{T}} \cdot \mathbf{e} \qquad (2.1.14)$$

It can be demonstrated that both KED and RK algorithms give exactly the same results (Stein, 1999; Hengl et al., 2007b). Start from KED where the predictions are made as in ordinary kriging using $\hat{z}_{\text{KED}}(\mathbf{s}_0) = \lambda_{\text{KED}}^{\mathbf{T}} \cdot \mathbf{z}$. The KED kriging weights $(\lambda_{\text{KED}}^{\mathbf{T}})$ are obtained by solving the system (Wackernagel, 2003, p.179):

$$\begin{bmatrix} \mathbf{C} & \mathbf{q} \\ \mathbf{q}^{\mathbf{T}} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \lambda_{\text{KED}} \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{q}_0 \end{bmatrix} \qquad (2.1.15)$$

where ϕ is a vector of Lagrange multipliers. Writing this out yields:

$$\mathbf{C} \cdot \lambda_{\text{KED}} + \mathbf{q} \cdot \phi = \mathbf{c}_0$$
$$\mathbf{q}^{\mathbf{T}} \cdot \lambda_{\text{KED}} = \mathbf{q}_0 \qquad (2.1.16)$$

from which follows:

$$\mathbf{q}^{\mathbf{T}} \cdot \lambda_{\text{KED}} = \mathbf{q}^{\mathbf{T}} \cdot \mathbf{C}^{-1} \cdot \mathbf{c}_0 - \mathbf{q}^{\mathbf{T}} \cdot \mathbf{C}^{-1} \cdot \mathbf{q} \cdot \phi \qquad (2.1.17)$$

and hence:

$$\phi = \left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{c_0} - \left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q_0} \qquad (2.1.18)$$

where the identity $\mathbf{q^T} \cdot \lambda_{\text{KED}} = \mathbf{q_0}$ has been used. Substituting ϕ back into Eq. (2.1.16) shows that the KED weights equal (Papritz and Stein, 1999, p.94):

$$\begin{aligned}
\lambda_{\text{KED}} &= \mathbf{C^{-1}} \cdot \mathbf{c_0} - \mathbf{C^{-1}} \cdot \mathbf{q} \cdot \left[\left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{c_0} - \left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q_0}\right] \\
&= \mathbf{C^{-1}} \cdot \left[\mathbf{c_0} + \mathbf{q} \cdot \left(\mathbf{q^T} \cdot \mathbf{C^{-1}}\mathbf{q} \cdot\right)^{-1} \cdot \left(\mathbf{q_0} - \mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{c_0}\right)\right]
\end{aligned}$$
$$(2.1.19)$$

Let us now turn to RK. Recall from Eq.(2.1.3) that the GLS estimate for the vector of regression coefficients is given by:

$$\hat{\beta}_{\text{GLS}} = \left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{z} \qquad (2.1.20)$$

and weights for residuals by:

$$\lambda_0^{\mathbf{T}} = \mathbf{c_0^T} \cdot \mathbf{C^{-1}} \qquad (2.1.21)$$

and substituting these in RK formula (Eq.2.1.4) gives:

$$\begin{aligned}
&= \mathbf{q_0^T} \cdot \hat{\beta}_{\text{GLS}} + \lambda_0^{\mathbf{T}} \cdot (\mathbf{z} - \mathbf{q} \cdot \hat{\beta}_{\text{GLS}}) \\
&= \left[\mathbf{q_0^T} \cdot \left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q^T} \cdot \mathbf{C^{-1}} + \mathbf{c_0^T} \cdot \mathbf{C^{-1}} - \mathbf{c_0^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q} \cdot \left(\mathbf{q^T} \cdot \mathbf{C^{-1}}\mathbf{q}\right)^{-1} \cdot \mathbf{q^T} \cdot \mathbf{C^{-1}}\right] \cdot \mathbf{z} \\
&= \mathbf{C^{-1}} \cdot \left[\mathbf{c_0^T} + \mathbf{q_0^T} \cdot \left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \mathbf{q^T} - \mathbf{c_0^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q} \cdot \left(\mathbf{q^T} \cdot \mathbf{C^{-1}}\mathbf{q}\right)^{-1} \cdot \mathbf{q^T}\right] \cdot \mathbf{z} \\
&= \mathbf{C^{-1}} \cdot \left[\mathbf{c_0} + \mathbf{q} \cdot \left(\mathbf{q^T} \cdot \mathbf{C^{-1}} \cdot \mathbf{q}\right)^{-1} \cdot \left(\mathbf{q_0} - \mathbf{q^T} \cdot \mathbf{C^{-1}}\mathbf{c_0}\right)\right] \cdot \mathbf{z}
\end{aligned}$$
$$(2.1.22)$$

The left part of the equation is equal to Eq.(2.1.19), which proves that KED will give the same predictions as RK if same inputs are used. A detailed comparison of RK and KED using a small dataset in MS Excel is also available as supplementary material.

Although the KED seems, at first glance, to be computationally more straightforward than RK, the variogram parameters for KED must also be estimated from regression residuals, thus requiring a separate regression modelling step. This regression should be GLS because of the likely spatial correlation between residuals. Note that many analyst use instead the OLS residuals, which may not be too different from the GLS residuals (Hengl et al., 2007b; Minasny and McBratney, 2007). However, they are not optimal if there is any spatial correlation, and indeed they may be quite different for clustered sample points or if the number of samples is relatively small ($\ll 200$).

A limitation of KED is the instability of the extended matrix in the case that the covariate does not vary smoothly in space (Goovaerts, 1997, p.195). RK has the advantage that it explicitly separates trend estimation from spatial prediction of residuals, allowing the use of arbitrarily-complex forms of regression, rather than the simple linear techniques that can be used with KED (Kanevski et al., 1997). In addition, it allows the separate interpretation of the two interpolated components. For these reasons the use of the term *regression-kriging* over *universal kriging* has been often advocated (Hengl et al., 2007b). The emphasis on regression is important also because fitting of the deterministic part of variation (regression) is often more beneficial for the quality of final maps than fitting of the stochastic part (residuals).

2.1.3 A simple example of regression-kriging

The next section illustrates how regression-kriging computations work and compares it to ordinary kriging using the textbook example from Burrough and McDonnell (1998, p.139-141), in which five measurements are used to predict a value of the target variable (z) at an unvisited location ($\mathbf{s}_0$) (Fig. 2.4a). We extend this example by adding a hypothetical auxiliary data source: a raster image (10×10 pixels) (Fig. 2.4b), which has been constructed to show a strong negative correlation with the target variable at the sample points.
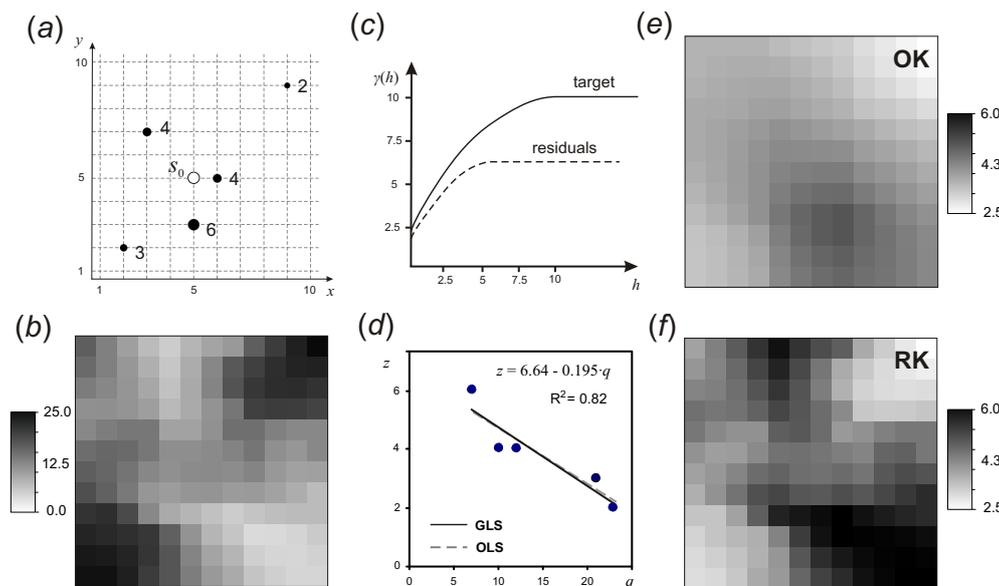

Fig. 2.4: Comparison of ordinary kriging and regression-kriging using a simple example with 5 points (Burrough and McDonnell, 1998, p.139–141): (a) location of the points and unvisited site; (b) values of the covariate q; (c) variogram for target and residuals, (d) OLS and GLS estimates of the regression model and results of prediction for a 10×10 grid using ordinary kriging (e) and regression-kriging (f). Note how the RK maps reflects the pattern of the covariate.

The RK predictions are computed as follows:

(1.) **Determine a linear model** of the variable as predicted by the auxiliary map q. In this case the correlation is high and negative with OLS coefficients b_0=6.64 and b_1=-0.195 (Fig. 2.4d).

(2.) **Derive the OLS residuals** at all sample locations as:

$$e^*(\mathbf{s}_i) = z(\mathbf{s}_i) - [b_0 + b_1 \cdot q(\mathbf{s}_i)] \tag{2.1.23}$$

For example, the point at (x=9, y=9) with z=2 has a prediction of $6.64 - 0.195 \cdot 23 = 1.836$, resulting in an OLS residual of $e^* = -0.164$.

(3.) **Model the covariance structure of the OLS residuals**. In this example the number of points is far too small to estimate the autocorrelation function, so we follow the original text in using a hypothetical variogram of the target variable (spherical model, nugget C_0=2.5, sill C_1=7.5 and range R=10) and residuals (spherical model, C_0=2, C_1=4.5, R=5). The residual model is derived from the target variable model of the text by assuming that the residual variogram has approximately the same form and nugget but a somewhat smaller sill and range (Fig. 2.4c), which is often found in practice (Hengl et al., 2004a).

(4.) **Estimate the GLS coefficients** using Eq. 2.1.3. In this case we get just slightly different coefficients b_0=6.68 and b_1=-0.199. The GLS coefficients will not differ much from the OLS coefficients as long there is no significant clustering of the sampling locations (Fig. 2.4d) as in this case.

(5.) **Derive the GLS residuals at all sample locations**:

$$e^{**}(\mathbf{s}_i) = z(\mathbf{s}_i) - [b_0 + b_1 \cdot q(\mathbf{s}_i)] \tag{2.1.24}$$

Note that the b now refer to the GLS coefficients.

(6.) **Model the covariance structure of the GLS residuals** as a variogram. In practice this will hardly differ from the covariance structure of the OLS residuals.

(7.) **Interpolate the GLS residuals using simple kriging** (SK) with known expected mean of the residuals (by definition 0) and the modelled variogram. In this case at the unvisited point location $(5,5)$ the interpolated residual is -0.081.

(8.) **Add the GLS surface to the interpolated GLS residuals** at each prediction point. At the unvisited point location $(5,5)$ the auxiliary variable has a value 12, so that the prediction is then:

$$\hat{z}(5,5) = b_0 + b_1 \cdot q_i + \sum_{i=1}^{n} \lambda_i(\mathbf{s}_0) \cdot e(\mathbf{s}_i)$$

$$\tag{2.1.25}$$

$$= 6.68 - 0.199 \cdot 12 - 0.081 = 4.21$$

which is, in this specific case, a slightly different result than that derived by OK with the hypothetical variogram of the target variable ($\hat{z}$=4.30).

The results of OK (Fig. 2.4e) and RK (Fig. 2.4f) over the entire spatial field are quite different in this case, because of the strong relation between the covariate and the samples. In the case of RK, most of variation in the target variable (82%) has been accounted for by the predictor. Unfortunately, this version of RK has not been implemented in any software package yet[3], which might change in the near future (see further §3.7.3).

2.2 Local versus localized models

In many geostatistical packages, a user can opt to limit the selection of points to determine the kriging weights by setting up a maximum distance and/or minimum and maximum number of point pairs (e.g. take only the closest 50 points). This way, the calculation of the new map can be significantly speed up. In fact, kriging in global neighbourhood where $n \gg 1000$ becomes cumbersome because of computation of $\mathbf{C}^{-1}$ (Eq.1.3.5). Recall from §1.3.1 that the importance of points (in the case of ordinary kriging and assuming a standard initial variogram model) exponentially decreases with their distance from the point of interest. Typically, geostatisticians suggest that already first 30–60 closest points will be good enough to obtain stable predictions.

---

[3]Almost all geostatistical packages implement the KED algorithm because it is mathematically more elegant and hence easier to program.
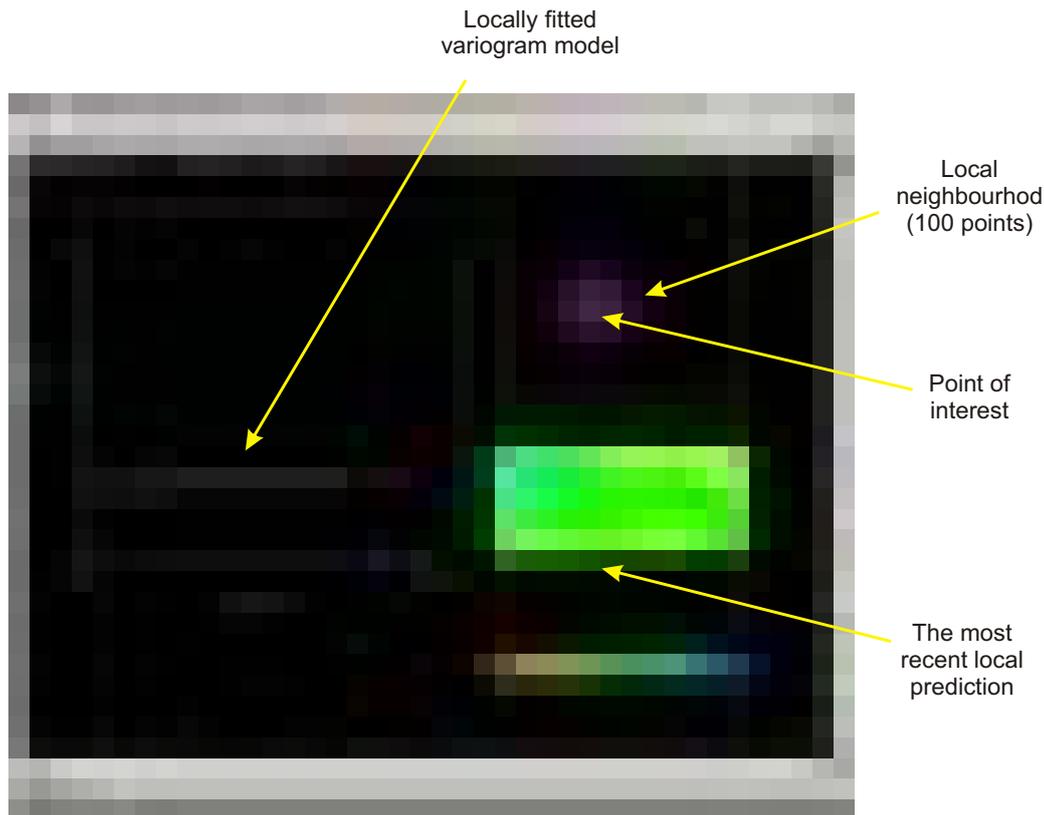
Fig. 2.5: Local variogram modelling and local ordinary kriging using a moving window algorithm in Vesper: a user can visually observe how the variograms change locally. Courtesy of Budiman Minasny.

A prediction model where the search radius for derivation of kriging weights (Eq.1.3.4) is limited to a local neighbourhood can be termed **localized prediction model**. There is a significant difference between *localized* and *local* prediction model, which often confuses inexperienced users. For example, if we set a search radius to re-estimate the variogram model, then we speak about a **local prediction model**, also known as the **moving window kriging** (Walter et al., 2001). The local prediction model assumes that the variograms (and regression models) are non-stationary, i.e. that they need to be estimated locally (Haas, 1990).

While localized prediction models are usually just a computational trick to speed up the calculations, local prediction models are computationally much more demanding. Typically, they need to allow automated variogram modelling and filtering of improbable models to prevent artefacts in the final outputs. A result of local prediction model (e.g. moving window variogram modelling) are not only maps of predictions, but also spatial distribution of the fitted variogram parameters (Fig. 2.6). This way we can observe how does the nugget variation changes locally, which parts of the area are smooth and which are noisy etc. Typically, local variogram modelling and prediction make sense only when we work with large point datasets (e.g. $\gg 10^3$ of field observations), which is still not easy to find. In addition, local variogram modelling is not implemented in many packages. In fact, the author is only aware of one: Vesper (Fig. 2.5).

In the case of regression-kriging, we could also run both localized and local models. This way we will not only produce maps of variogram parameters but we would also be able to map the regression coefficients. In the case of kriging with external drift, some
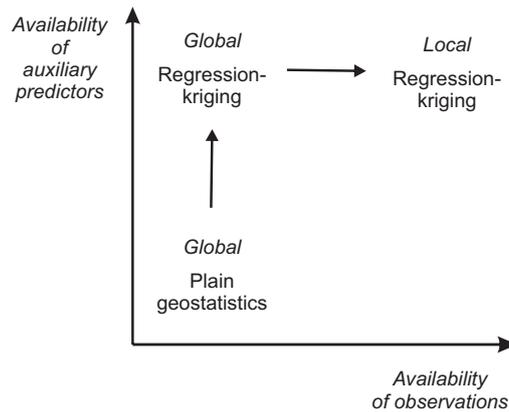
Fig. 2.6: Local regression-kriging is a further sophistication of regression-kriging. It will largely depend on the availability of auxiliary and field data.

users often make a mistake[4] and limit the search window to speed up the calculations. This is obviously a simplification, because in the case of KED both regression and kriging part of predictions are solved at the same time. Hence, if we limit the search window, but keep a constant variogram model, we could obtain very different predictions then if we use a global (regression-kriging) model. Only if the variogram of residuals if absolutely stationary, then we can limit the search window to fit the KED weights. In practice, only either global (constant variogram) or local prediction models (locally estimated regression models and variograms of residuals) should be used for KED fitting.

2.3   Spatial prediction of categorical variables

Although geostatistics is primarily intended for use with continuous environmental variables, it is also fit for use with various types of categorical or class-type variables. Geostatistical analysis of categorical variables is by many referred to as the **indicator geostatistics** (Bierkens and Burrough, 1993). In practice, indicator kriging leads to many computational problems, which probably explains why there are not many operational applications of geostatistical mapping of categorical variables in the world (Hession et al., 2006). For example, it will typically be difficult to fit variogram for less frequent classes that occur at isolated classes (Fig. 2.7d).

Let us denote the field observations of a class-type variable as $z_c(\mathbf{s}_1), z_c(\mathbf{s}_2), ..., z_c(\mathbf{s}_n)$, where $c_1, c_2, ..., c_k$ are discrete categories (or states) and k is the total number of classes. A technique that estimates the soil-classes at new unvisited location $\hat{z}_c(\mathbf{s}_0)$, given the input point dataset $(z_c(\mathbf{s}_1), z_c(\mathbf{s}_2), ..., z_c(\mathbf{s}_n))$, can then be named a class-type interpolator. If spatially exhaustive predictors $q_1, q_2, ..., q_p$ (where p is the number of predictors) are available, they can be used to map each category over the area of interest. So far, there is a limited number of techniques that can achieve this:

Multi-indicator co-kriging — The simple multi-indicator kriging can also be extended to a case where several covariates are used to improve the predictions. This technique is known by the name *indicator* (soft) *co-kriging* (Journel, 1986). Although the mathematical theory is well explained (Bierkens and Burrough, 1993;

[4]This is probably a mistake of the software that allows it.
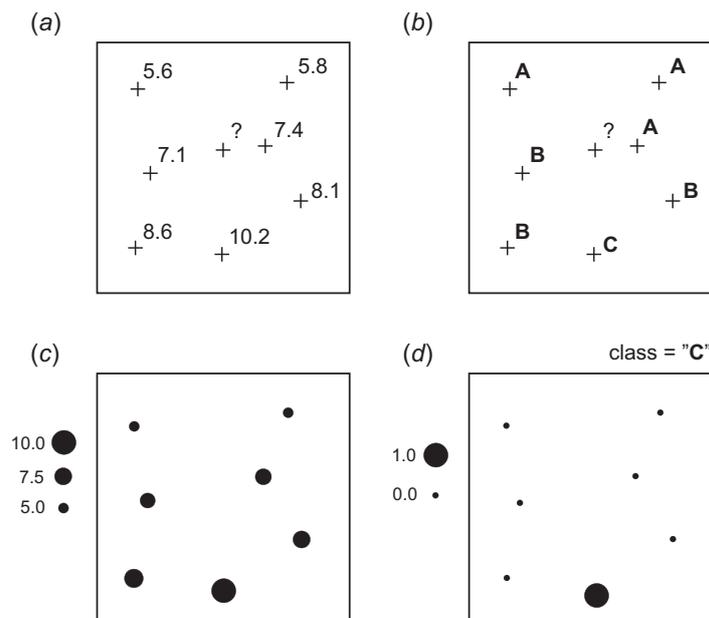
Fig. 2.7: Difficulties of predicting point-class data (b) and (d), as compared to quantitative variables (a) and (c), is that the class-interpolators are typically more complex and computationally more time-consuming.

Goovaerts, 1997; Pardo-Iguzquiza and Dowd, 2005), the application is cumbersome because of the need to fit a very large number of cross-covariance functions.

Multinomial Log-linear regression — This a generalization of logistic regression for situations when there are multiple classes of a target variable. Each class gets a separate set of regression coefficients ($\beta_{\mathbf{c}}$). Because the observed values equal either 0 or 1, the regression coefficients need to be solved through a maximum likelihood iterative algorithm (Bailey et al., 2003), which makes the whole method somewhat more computationally demanding than simple multiple regression.

Regression-kriging of indicators — One approach to interpolate soil categorical variables is to first assign memberships to point observations and then to interpolate each membership separately. This approach was first elaborated by de Gruijter et al. (1997) and then applied by Bragato (2004) and Triantafilis et al. (2001). An alternative is to first map cheap, yet descriptive, diagnostic distances and then classify these per pixel in a GIS (Carré and Girard, 2002).

In the case of logistic regression, the odds to observe a class (c) at new locations are computed as:

$$\hat{z}_c^+(\mathbf{s}_0) = \left[1 + \exp\left(-\beta_{\mathbf{c}}{}^{\mathbf{T}} \cdot \mathbf{q_0}\right)\right]^{-1}; \ \ c = 1, 2, .., k \qquad (2.3.1)$$

where $\hat{z}_c^+(\mathbf{s}_0)$ are the estimated odds for class (c) at new location s_0 and k is the number of classes. The multinomial logistic regression can also be extended to regression-kriging (for a complete derivation see Hengl et al. (2007c)). This means that the regression modelling is supplemented with the modelling of variograms for regression residuals, which can then be interpolated and added back to the regression estimate. So the

predictions are obtained using:

$$\hat{z}_c^+(\mathbf{s}_0) = \left[1 + \exp\left(-\beta_{\mathbf{c}}^{\mathbf{T}} \cdot \mathbf{q_0}\right)\right]^{-1} + \hat{e}_c^+(\mathbf{s}_0) \tag{2.3.2}$$

where $\hat{e}_c^+$ are the interpolated residuals. The extension from multinomial regression to regression-kriging is not as simple as it seems. This is because the estimated values at new locations in Eq.(2.3.2) are constrained within the indicator range, which means that interpolation of residuals might lead to values outside the physical range (< 0 or > 1). A solution to this problem is to, instead of using the (crisp) indicator values, work with (continuous) memberships (μ_c). Memberships are more suitable both for regression and geostatistical modelling, which has been confirmed by several authors (McBratney et al., 1992; de Gruijter et al., 1997; Triantafilis et al., 2001). Memberships can be directly linearized using the logit transformation:

$$\mu_c^+ = \ln\left(\frac{\mu_c}{1 - \mu_c}\right); \qquad 0 < \mu_c < 1 \tag{2.3.3}$$

where μ_c are the membership values used as input to interpolation. Then, all fitted values will be within the physical range (0–1). The predictions of memberships for class c at new locations are then obtained using the standard regression-kriging model:

$$\hat{\mu}_c^+(\mathbf{s}_0) = \mathbf{q_0^T} \cdot \hat{\beta}_{\mathbf{c},\text{GLS}} + \lambda_{\mathbf{c},\mathbf{0}}^{\mathbf{T}} \cdot \left(\mu_{\mathbf{c}}^+ - \mathbf{q} \cdot \hat{\beta}_{\mathbf{c},\text{GLS}}\right) \tag{2.3.4}$$

The interpolated values can then be back-transformed to the membership range using (Neter et al., 1996):

$$\hat{\mu}_c(\mathbf{s}_0) = \frac{e^{\hat{\mu}_c^+(\mathbf{s}_0)}}{1 + e^{\hat{\mu}_c^+(\mathbf{s}_0)}} \tag{2.3.5}$$

In the case of regression-kriging of memberships, both spatial dependence and correlation with the predictors are modelled in a statistically sophisticated way. In addition, regression-kriging of memberships allows fitting of each class separately, which facilitates the understanding of the distribution of soil variables and the identification of problematic classes, i.e. classes which are not correlated with the predictors or do not show any spatial autocorrelation.

Spatial prediction of memberships can be excessive in computation time. Another problem is that, if the interpolated classes (odds, memberships) are fitted only by using the sampled data, the predictions of the odds/memberships will commonly not sum to unity at new locations. In this case, we needed to standardized values for each grid node by diving the original values by the sum of odds/memberships to ensure that they sum to unity, which is a short-cut solution. Obviously, an algorithm, such as compositional regression-kriging[5] will need to be developed.

A number of alternative hybrid class-interpolators exists, e.g. the Bayesian Maximum Entropy (BME) approach by D'Or and Bogaert (2005). Another option is to use Markov-chain algorithms (Li et al., 2004, 2005). However, note that although use of the BME and Markov-chain type of algorithms is a promising development, its computational complexity makes it still far from use in operational mapping.

---

[5]Walvoort and de Gruijter (2001), for example, already developed a compositional solution for ordinary kriging that will enforce estimated values to sum to unity at all locations.

2.4 Geostatistical simulations

Regression-kriging can also be used to generate simulations of a target variable using the same inputs as in the case of spatial prediction system. An equiprobable realisation of an environmental variable can be generated by using the sampled values and their variogram model:

$$Z^{(\texttt{SIM})}(\mathbf{s}_0) = E\left\{Z|z(\mathbf{s}_j), \gamma(\mathbf{h})\right\} \tag{2.4.1}$$

where $Z^{(\texttt{SIM})}$ is the simulated value at the new location. The most common technique in geostatistics that can be used to generate equiprobable realisations is the **Sequential Gaussian Simulation** (Goovaerts, 1997, p.380-392). It starts by defining a random path for visiting each node of the grid once. At first node, kriging is used to determine the location-specific mean and variance of the conditional cumulative distribution function. A simulated value can then be drawn by using the inverse normal distribution (Banks, 1998):

$$z_i^{\texttt{SIM}} = \hat{z}_i + \hat{\sigma}_i \cdot \sqrt{-2 \cdot \ln(1-A)} \cdot \cos(2 \cdot \pi \cdot B) \tag{2.4.2}$$

where $z_i^{\texttt{SIM}}$ is the simulated value of the target variable with induced error, A and B are the independent random numbers within the $0 - 0.99\ldots$ range, $\hat{z}_i$ is the estimated value at ith location, and $\hat{\sigma}_i$ is the regression-kriging error. The simulated value is then added to the original dataset and the procedure is repeated until all nodes have been visited. Geostatistical simulations are used in many different fields to generate multiple realisations of the same feature (Heuvelink, 1998; Kyriakidis et al., 1999), or to generate realistic visualizations of a natural phenomena (Hengl and Toomanian, 2006; Pebesma et al., 2007).

2.5 Spatio-temporal regression-kriging

The 2D space models can be extended to the time domain, which leads to **spatio-temporal geostatistics** (Kyriakidis and Journel, 1999). The universal kriging model (Eq.2.1.1) then modifies to:

$$Z(\mathbf{s}, t) = m(\mathbf{s}, t) + \varepsilon'(\mathbf{s}, t) + \varepsilon'' \tag{2.5.1}$$

where $\varepsilon'(\mathbf{s}, t)$ is the spatio-temporally autocorrelated residual.

In practice, spatio-temporal interpolation follows the geostastical interpolation principle as explained in Eq.(1.1.2), except that here the variograms are estimated in three dimensions (two-dimensional position x and y and '*position*' in time). From the mathematical aspect, the extension from the static 2D interpolation to the 3D interpolation is then rather simple. Regression modelling can be simply extended to a space-time model by adding time as a predictor. For example, a spatio-temporal regression model for interpolation of mean-daily land surface temperature (see further §2.7.2) would look like this:

$$
\begin{aligned}
LST(\mathbf{s}_0, t_0) = {} & b_0 + b_1 \cdot DEM(\mathbf{s}_0) + b_2 \cdot LAT(\mathbf{s}_0) + b_3 \cdot DISTC(\mathbf{s}_0) \\
& + b_4 \cdot SOLAR(\mathbf{s}_0, t_0) + b_5 \cdot \cos\left([t_0 - \phi] \cdot \frac{\pi}{180}\right); \qquad \Delta t = 1 \text{ day}
\end{aligned}
\tag{2.5.2}
$$

where DEM is the elevation map, LAT is the map showing distance from the equator, $DISTC$ is the distance from the coast line, $SOLAR$ is the direct solar insolation for

a given cumulative Julian day $t \in (0, +\infty)$, $\cos(t)$ is a generic function to account for seasonal variation of values and ϕ is the phase angle[6]. *DEM*, *LAT*, *DISTC* are temporally-constant predictors, while solar insolation maps need to be provided for each time interval used for data fitting. The residuals from this regression model can then be analysed for (spatio-temporal) auto-correlation. In gstat, extension from 2D to 3D variograms is possible by extending the variogram parameters: for 3D space-time variograms five values should be given in the form `anis = c(p,q,r,s,t)`, where `p` is the angle for the **principal direction of continuity** (measured in degrees, clockwise from y, in direction of x), `q` is the **dip angle** for the principal direction of continuity (measured in positive degrees up from horizontal), `r` is the third rotation angle to rotate the two minor directions around the principal direction defined by `p` and `q`. A positive angle acts counter-clockwise while looking in the principal direction.

Once we have fitted the space-time variogram, we can then run regression-kriging to estimate the values at 3D locations. In practice, we only wish to produce maps for a given time interval (t_0=constant), i.e. to produce 2D-slices of values in time (Fig. 2.8).
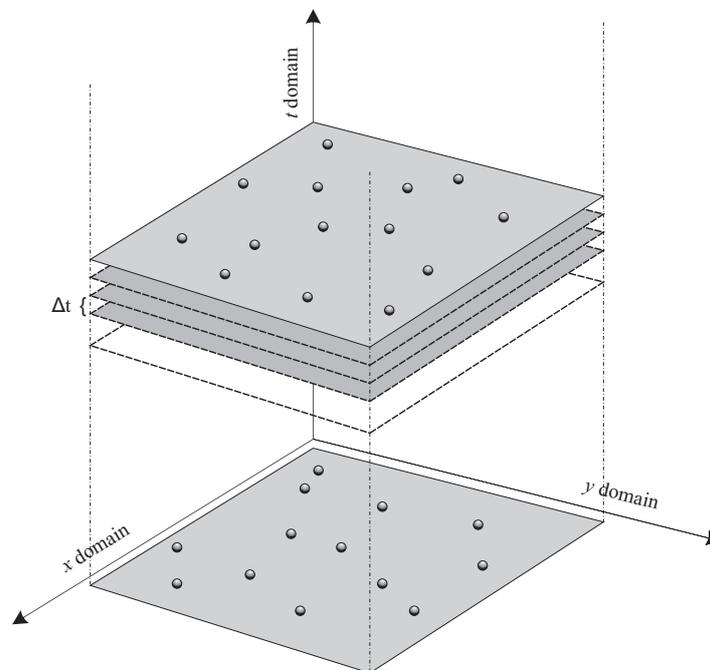

Fig. 2.8: Extension of a 2D prediction model to the space-time domain. Note that in the space-time cube, the amount of pixels *in play* exponentially increases as a function of: width × height × number of predictors × number of time intervals.

Note that, in order to yield accurate predictions using spatio-temporal techniques, dense sampling in both space and time is required. This means that existing natural resource surveys that have little to no repetition in time ($\ll$10 repetitions in time) cannot be adopted. Not to mention the computational complexity as the maps of predictors now multiply by the amount of time intervals. In addition, estimation of the spatio-temporal variograms will often be a cumbersome because we need to fit space-time models, for which we might not have enough space-time observations.

---

[6]A time delay from the coldest day.

A specific extension of the general model from Eq.(2.5.1) is to estimate the deterministic part of variation by using process-based (simulation) models. In this case an environmental variable is predicted from a set of environmental predictors incorporated in a dynamic model (Eq.1.3.12):

$$Z(\mathbf{s}, t) = f_{s,c,r,p,a}(t) + \varepsilon'(\mathbf{s}, t) + \varepsilon'' \tag{2.5.3}$$

where s, c, r, p, a are the input (zero-stage) environmental conditions and f is a mathematical deterministic function that can be used to predict the values for a given space-time position. This can be connected with the Einstein's assumption that the Universe is in fact a trivial system that can be modelled and analysed using a one–dimensional differential equation — in which everything is a function of time[7]. Some examples of operational soil-landscape process-based models are given by Minasny and McBratney (2001) and Schoorl et al. (2002). In vegetation science, for example, global modelling has proven to be very efficient for explanation of the actual distribution of vegetation and of global changes (Bonan et al., 2003). Integration of environmental process-based models will soon lead to development of a global dynamic model of environmental systems that would then 'feed' different applications/national systems.

2.6 Sampling strategies and optimisation algorithms

Understanding the concepts of regression-kriging is not only important to know how to generate maps, but also to know how to prepare a sampling plan and eventually minimize the survey costs. Because the costs of the field survey are usually the biggest part of the survey budget, this issue will become more and more important in the coming years.

So far, two main groups of sampling strategies have been commonly utilized for the purpose of environmental mapping:

- **Regular sampling** — This has the advantage that it systematically covers the area of interest (maximized mean shortest distance), so that the overall prediction variance is usually minimized[8]. The disadvantage of this technique is that it misrepresents distances smaller than the grid size (short range variation).

- **Randomized sampling** — This has the advantage that it represents all distances between the points, which is beneficial for the variogram estimation. The disadvantage is that the spreading of the points in geographic space is lower than in the case of regular sampling, so that the overall precision of the final maps will often be lower.

None of two strategies is universally applicable so that often their combination is recommended: e.g. put half of the points using regular and half using a randomized strategy. Both sampling strategies belong to the group of design-based sampling. A difference between a design-based sampling (e.g. simple random sampling) and the model-based design is that, in the case of the model-based design, the model is defined and commonly a single optimal design that maximizes/minimizes some criteria can be produced.

In the case of regression-kriging, there are much more possibilities to improve sampling than by using design-based sampling. First, in the case of preparing a sampling

---

[7]James Peebles, Princeton, 1990; published in "God's Equation: Einstein, Relativity, and the Expanding Universe" by Amir D. Aczel.

[8]If ordinary kriging is used to generate predictions.
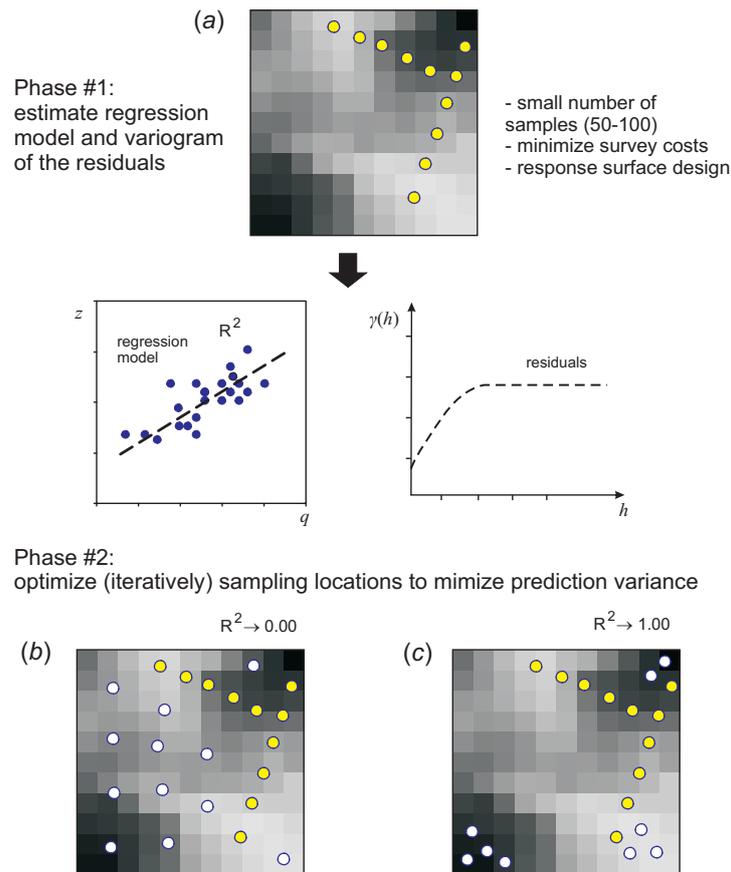
Fig. 2.9: Principle of the two-phase model-based sampling based on the regression-kriging model: (a) in the first phase, sampling aims at estimating the regression-kriging model; (b) if the correlation with predictors is low, the additional sampling will probably lead to higher spreading in the geographical space; (c) if the correlation with predictors is high, then the sampling will probably only follow the extremes of the features space.

design for new survey, the samples can be more objectively located by using some **response surface design** (Hengl et al., 2004b), including the **Latin hypercube sampling** (Minasny and McBratney, 2006). The Latin hypercube sampling will ensure that all points are well-placed in the feature space defined by the environmental factors — these will later be used as predictors — and that the extrapolation in feature space is minimized. Second, once we have collected samples and estimated the regression-kriging model, we can now optimize sampling and derive (1) number of required additional observations and (2) their optimal location in both respective spaces. This leads to a principle of the two-phase[9] model-based sampling (Fig. 2.9).

The **two-phase sampling** is a guarantee of minimization of the survey costs. In the first phase, the surveyors will produce a sampling plan with minimum survey costs — just to have enough points to get a 'rough' estimate of the regression-kriging model. Once the model is approximated (correlation and variogram model), and depending on the prescribed accuracy (overall prediction variance), the second (additional) sampling

---

[9]Ideally, already one iteration of additional sampling should guarantee map of required accuracy/quality. In practice, also the estimation of model will need to be updated with additional predictors, so that one might need to run several sampling iterations.

plan can be generated. Now we can re-estimate the regression-kriging model and update the predictions so that they fit exactly our prescribed precision requirements. Brus and Heuvelink (2007) recently tested the use of simulated annealing to produce optimal designs based on the regression-kriging model and concluded that the resulting sampling plans will lead to hybrid patterns, well spread in both feature and geographical space. Such algorithms are unfortunately still not available to a wider community (see also §2.8.3).

Smarter allocation of the points in the feature and geographic space often proves that equally precise maps could have been produced with much less points than actually collected (see further §4.7.2). This might surprise you, but it has a strong theoretical background. Especially if the predictors are highly correlated with the target variable and if this correlation is close to linear, there is really no need to collect many samples in the study area (see e.g. Fig. 2.11). In order to produce precise maps, it would be enough if we spread them around extremes[10] of the feature space and possibly maximized their spreading in the area of interest. Of course, number of sampling points is mainly dictated by our precision requirements, so that more accurate (low overall precision variance) and detailed (fine cell size) maps of environmental variables will often require denser sampling densities.

2.7 Fields of application

With the rapid development of remote sensing and geoinformation science, natural resources survey teams are now increasingly creating their products (geoinformation) using ancillary data sources and computer programs — the so-called *direct-to-digital* approach. For example, sampled concentrations of heavy metals can be mapped with higher accuracy/detail if information about the sources of pollution (distance to industrial areas and traffic or map showing the flooding potential) is used. In the following sections, a short review of the groups of application where regression-kriging has shown its potential will be given.

2.7.1 Soil mapping applications

In digital soil mapping, soil variables such as pH, clay content or concentration of a heavy metal, are increasingly mapped using the regression-kriging framework: the deterministic part of variation is dealt with maps of soil forming factors (climatic, relief-based and geological factors) and the residuals are dealt with kriging (McBratney et al., 2003). The same techniques is now used to map categorical variables (Hengl et al., 2007c). A typical soil mapping project based on geostatistics will also be demonstrated in the following chapter of this handbook. This follows the generic framework for spatial prediction set in Hengl et al. (2004a) and applicable also to other environmental and geosciences (Fig. 2.10).

In geomorphometry, auxiliary maps, such as maps of drainage patterns, land cover and remote sensing-based indices, are increasingly used for geostatistical modelling of topography together with point datasets. Auxiliary maps can help explain spatial distribution of errors in DEMs and regression-kriging can be used to generate equiprobable realisations of topography or map the errors in the area of interest (Hengl et al., 2007a). Such hybrid geostatistical techniques will be more and more attractive for handling rich LiDAR and radar-based topographic data, both to analyse their inherent geostatistical

---

[10]In statistics, these are referred to as the D-designs. For more info see Hengl et al. (2004b).
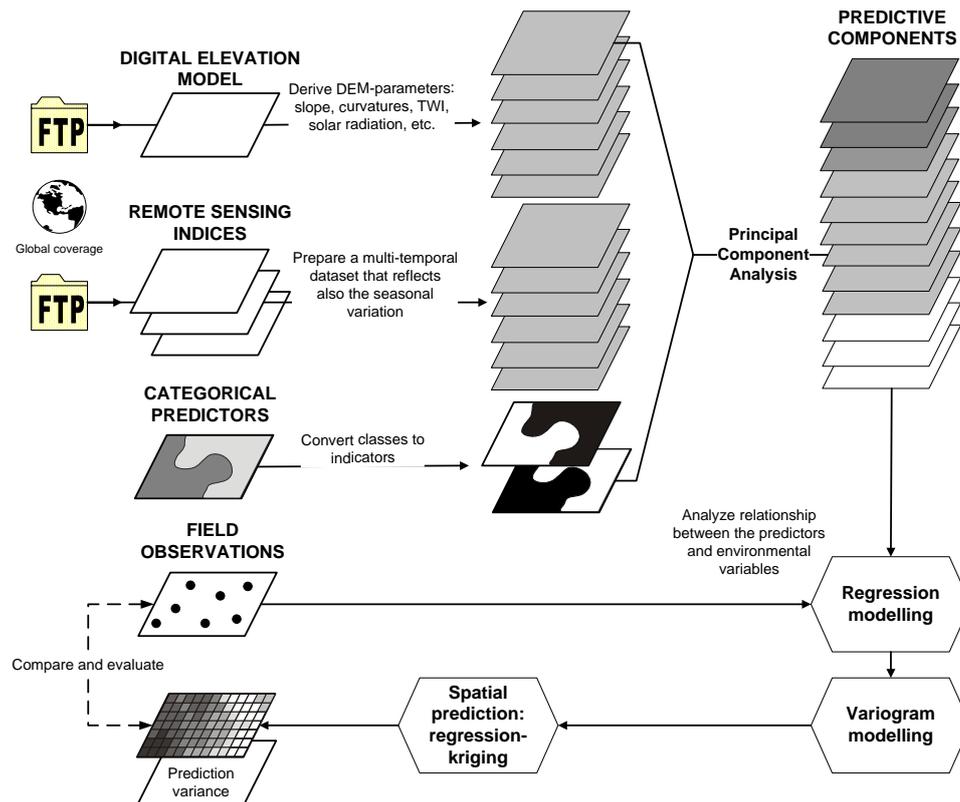
Fig. 2.10: A generic framework for digital soil mapping based on regression-kriging.

properties and generate DEMs fit-for-use in various environmental and earth science applications.

2.7.2   Interpolation of climatic and meteorological data

Regression-kriging of climatic variables, especially the ones derived from DEMs, is now favoured in many climatologic applications (Jarvis and Stuart, 2001; Lloyd, 2005). DEMs are most commonly used to adjust measurements at meteorological stations to local topographic conditions. Other auxiliary predictors used range from distance to sea, meteorological images of land surface temperature, water vapor, short-wave radiation flux, surface albedo, snow Cover, fraction of vegetation cover (Table 3.2). In many cases, real deterministic models can be used to make predictions, so that regression-kriging is only used to calibrate the values using the real observations (D'Agostino and Zelenka, 1992, see also Fig. 2.3). An example in Fig. 2.11 demonstrates the benefits of using the auxiliary predictors to map climatic variables. In this case the predictors explained over 90% of variation in the land surface temperatures measured at 152 stations[11]. Such high R-square allows us to extrapolate the values much further from the original sampling locations, which would be completely inappropriate to do by using ordinary kriging. Note also that the range of values is now considerably larger than in the original data — temperatures range from -1 to 15°C, compared to 7–14°C range. The increase of the predictive capabilities using the auxiliary information and regression-kriging has been also reported by several participants of the recent Conference on spatial interpolation

---

[11]Meteorological and Hydrological service of Croatia: 1961–1990 climate normals.

in climatology and meteorology (Szalai et al., 2007).
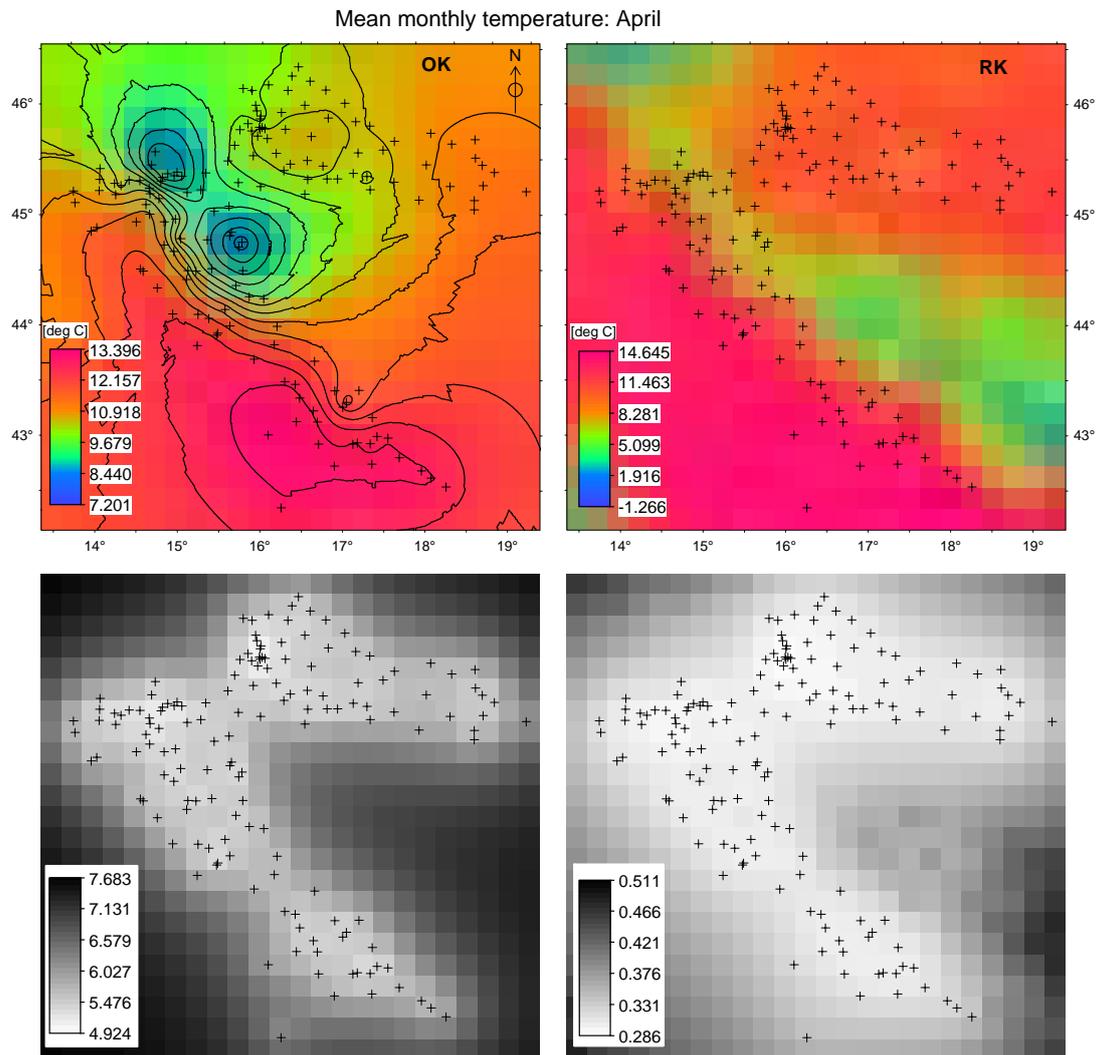
Fig. 2.11: The long-term mean monthly temperature for April interpolated using OK (left) and RK (right) at 1 km grid. Predictions (above) and prediction variances (below) in this case differ significantly. In this case, auxiliary predictors (elevation, latitude, direct annual solar insolation, distance from the coast line) explain 95.2% of variation in the original data (152 meteorological stations), hence the mapping precision is considerably higher for RK.

Interpolation of climatic and meteorological data is also interesting because the auxiliary (meteorological images) data are today increasingly collected in shorter time intervals so that time-series of images are available and can be used to develop spatio-temporal regression-kriging models. Note also that many meteorological prediction models can generate maps of forecasted conditions in the close-future time, which could then again be calibrated using the actual measurements and RK framework (Fig. 2.3).

2.7.3   Mapping plant and animal species

As mentioned previously in §1.1.1, geostatistical modelling of plant and animal data is somewhat more complicated because we commonly deal with dynamic and discrete

features. Nevertheless, RK and auxiliary environmental predictors can also be used to map such features over the area of interest. This is more a question of using an appropriate prediction models, such as the logistic regression-kriging model (Latimer et al., 2004). Pebesma et al. (2005), for example, used RK to interpolate bird densities over the North Sea, while Shamoun et al. (2005) produced 3D maps (slices) of bird densities over the land.

In summary, there is a distinct difference between field observation of animal and plant species and measurements of soil or meteorological variables. Especially the observations of animal species asks for high sampling densities in temporal dimension. However, if biological species are represented with quantitative composite measures (density, occurrence, biomass, habitat category), such measures are fit for use with standard spatio-temporal geostatistical tools.

2.8 Final notes about regression-kriging

At the moment, there are not many contra-arguments not to replace the existing traditional soil, vegetation, climatic, geological and similar maps with the maps produced using analytical techniques. Note that this does not mean that we should abandon the traditional concepts of field survey and that surveyors are becoming obsolete. On the contrary, surveyors continue to be needed to prepare and collect the input data and to assess the results of spatial prediction. On the other hand, they are less and less involved in the actual delineation of features or derivation of predictions, which is increasingly the role of the predictive models.

One such linear prediction techniques that is especially promoted in this handbook is regression-kriging (RK). It can be used to interpolate sampled environmental variables (both continuous and categorical) from large point sets. However, in spite of this and other attractive properties of RK, it is not as widely used in geosciences as might be expected. The barriers to widespread routine use of RK in environmental modelling and mapping are as follows. First, the statistical analysis in the case of RK is more sophisticated than for simple mechanistic or kriging techniques. Second, RK is computationally demanding[12] and often can not be run on standard PCs. The third problem is that many users are confused by the quantity of spatial prediction options, so that they are never sure which one is the most appropriate. In addition, there is a lack of user-friendly GIS environments to run RK. This is because, for many years GIS technologies and geostatistical techniques have been developing independently. Today, a border line between statistical and geographical computing is definitively fading away (as you will be more convinced in the last chapter of this guide).

2.8.1 Alternatives to RK

The competitors to RK include completely different methods that may fit certain situations better. If the auxiliary data is of different origin and reliability, the Bayesian Maximum Entropy approach might be a better alternative (D'Or, 2003). There are also machine learning techniques that combine neural network algorithms and robust prediction techniques (Kanevski et al., 1997). Henderson et al. (2005) used decision trees to predict various soil parameters from large quantity of soil profile data and with the help of land surface and remote sensing attributes. This technique is flexible, optimizes

---

[12]Why does RK takes so much time? The most enduring computations are connected with derivation of distances from the new point to all sampled points. This can be speed up by setting up a smaller search radius.

local fits and can be used within a GIS. However, it is statistically suboptimal because it ignores spatial location of points during the derivation of classification trees. The same authors (Henderson et al., 2005, pp.394–396) further reported that, although there is still some spatial correlation in the residuals, it is not clear how to employ it.

Regression-kriging must also be compared with alternative kriging techniques, such as **collocated co-kriging**, which also makes use of the auxiliary information. However, collocated co-kriging is developed for situations in which the auxiliary information is not spatially exhaustive (Knotters et al., 1995). CK also requires simultaneous modelling of both direct and cross-variograms, which can be time-consuming for large number of covariates[13]. In the case where the covariates are available as complete maps, RK will generally be preferred over CK, although CK may in some circumstances give superior results (D'Agostino and Zelenka, 1992; Goovaerts, 1999; Rossiter, 2007b). In the case auxiliary point samples of covariates, in addition to auxiliary raster maps, are available, regression-kriging can be combined with co-kriging: first the deterministic part can be dealt with the regression, then the residuals can be interpolated using co-kriging (auxiliary point samples) and added back to the estimated deterministic part of variation.

2.8.2   Limitations of RK

RK have shown a potential to become the most popular mapping technique used by environmental scientists because it is (a) easy to use, and (b) it outperforms plain geostatistical techniques. However, success of RK largely depends on characteristics of the case study i.e. quality of the input data. These are some main consideration one should have in mind when using RK:

(1.) *Data quality*: RK relies completely on the quality of data. If the data comes from different sources and have been sampled using biased or unrepresentative design, the predictions might be even worse than with simple mechanistic prediction techniques. Even a single bad data point can make any regression arbitrarily bad, which affects the RK prediction over the whole area.

(2.) *Under-sampling*: For regression modelling, the multivariate feature space must be well-represented in all dimensions. For variogram modelling, an adequate number of point-pairs must be available at various spacings. Webster and Oliver (2001, p.85) recommend at least 50 and preferably 300 points for variogram estimation. Neter et al. (1996) recommends at least 10 observations per predictor for multiple regression. We strongly recommend using RK only for data sets with more than 50 total observations and at least 10 observations per predictor to prevent over-fitting.

(3.) *Reliable estimation of the covariance/regression model*: The major dissatisfaction of using KED or RK is that both the regression model parameters and covariance function parameters need to be estimated simultaneously. However, in order to estimate coefficients we need to know covariance function of residuals, which can only be estimated after the coefficients (the chicken-egg problem). Here, we have assumed that a single iteration is a satisfactory solution, although someone might also look for other iterative solutions (Kitanidis, 1994). Lark et al. (2005) recently suggested that an iterative Restricted Maximum Likelihood (REML) approach should be used to provide an unbiased estimate of the variogram and regression

---

[13]Co-kriging requires estimation of $p + 1$ variograms, plus $[p \cdot (p + 1)] / 2$ cross-variograms, where the p is the number of predictors (Knotters et al., 1995).

coefficients. However, this approach is rather demanding for $\gg 10^3$ point data sets because for each iteration, an $n \times n$ matrix is inverted.

(4.) *Extrapolation outside the sampled feature space*: If the points do not represent feature space or represent only the central part of it, this will often lead to poor estimation of the model and poor spatial prediction. For this reason, it is important that the points be well spread at the edges of the feature space and that they be symmetrically spread around the center of the feature space (Hengl et al., 2004b). Assessing the extrapolation in feature space is also interesting to allocate additional point samples that can be used to improve the existing prediction models. This also justifies use of multiple predictors to fit the target variable, instead of using only the most significant predictor or first principal component, which if, for example, advocated by the Isatis development team (Bleines et al., 2004).

(5.) *Predictors with uneven relation to the target variable*: Auxiliary maps should have a constant physical relationship with the target variable in all parts of the study area, otherwise artefacts will be produced. An example is a single NDVI as a predictor of topsoil organic matter. If an agricultural field has just been harvested (low NDVI), the prediction map will (incorrectly) show very low organic matter content within the crop field.

(6.) *Intermediate-scale modelling*: RK has not been adapted to fit data locally, with arbitrary neighbourhoods for the regression as can be done with kriging with moving window (Walter et al., 2001). Many practitioners would like to adjust the neighbourhood to fit their concepts of the scale of processes that are not truly global (across the whole study area) but not fully local either.

(7.) *Data over-fitting problems*: Care needs to be taken when fitting the statistical models — today, complex models and large quantities of predictors can be used so that the model can fit the data almost 100%. But there is a distinction between the goodness of fit and true success of prediction that can not really be assessed without independent validation (Rykiel, 1996).

If any of these problems occur, RK can give even worse results than even non-statistical, empirical spatial predictors such as inverse distance interpolation or expert systems. The difficulties listed above might also be considered as challenges for the geostatisticians.

2.8.3   Beyond RK

Although the bibliometric research of Zhou et al. (2007) indicates that the field of geostatistics has already reached its peak in 1996–1998, the development of regression-kriging and similar hybrid techniques is certainly not over and the methods will continue to evolve both from theoretical and practical aspect. What you can certainly anticipate in the near future are the following five developments:

- *More sophisticated prediction models*: Typically, regression-kriging is sensitive to blunders in data, local outliers and small size datasets. To avoid such problems, we will experience an evolution of methods that are more generic and more robust to be used to any type of dataset. Recently, several authors suggested ways to make more sophisticated, more universally applicable BLUPs (Lark et al., 2005; Minasny and McBratney, 2007). We can anticipate a further development of

intelligent, iterative data fitting algorithms that can account for problems of local hot-spots, mixed data and poor sampling strategies.

- *Local regression-kriging*: As mentioned previously in §2.2, local regression-kriging algorithms are yet to be developed. Integration of the local prediction algorithms (Haas, 1990; Walter et al., 2001) would open many new data analysis possibilities. For example, with local estimation of the regression coefficients and variogram parameters, a user will be able to analyse which predictors are more dominant in different parts of the study area, and how much these parameters vary in space. The output of the interpolation with not be only a map of predictions, but also the maps of (local) regression coefficients, R-square, variogram parameters and similar.

- *User-friendly sampling optimisation packages*: Although methodologies both to plan a new sampling design and to optimize additional sampling designs have already been proposed (Minasny and McBratney, 2006; Brus and Heuvelink, 2007), neither simulated annealing neither Latin hypercube sampling are available for operational mapping. Development of user-friendly sampling design packages would make possible to generate (*smart*) sampling schemes at the click of button.

- *Intelligent data analysis reports generation*: The next generation of geostatistical packages will be intelligent. It will not only generate the predictions and prediction variances, but will also provide interpretation of the fitted models and analysis of the intrinsic properties of the input data sets. This will include detection of possible outliers and hot-spots, robust estimation of the non-linear regression model, assessment of the quality of the input data sets and final maps. I imagine that the system will need to be organized in a step-by-step processing wizards, which means that the algorithm will iteratively question the user about the reliability of specific points until the most suitable prediction model is applied.

- *Multi-temporal, multi-variate prediction models*: At the moment, most of the geostatistical mapping projects in environmental sciences focus on mapping a single variable sampled in a short(er) period of time and for a local area of interest. It will not take too long until we will have a global repository of (multi-temporal) predictors (see e.g. Table 3.2) and point data sets that could then be interpolated all at once (to employ all possible relationships and cross-correlations). The future data sets will definitively be multi-temporal and multi-variate, and it will certainly ask for more powerful computers and more sophisticated spatio-temporal 3D mapping tools. Consequently, outputs of the spatial prediction models will be animations and multimedia, rather then simple and static 2D maps.

Although we can observe that with the more sophisticated methods (e.g. REML approach), we are able to produce more realistic models, the quality of the output maps depends much more on the quality of input data (Minasny and McBratney, 2007). Hence, we can also anticipate that evolution of technology such as hyperspectral remote sensing and LiDAR will contribute to the field of geostatistical mapping even more than the development of the more sophisticated algorithms.

Finally, we can conclude that an unavoidable trend in the evolution of spatial prediction models will be a **development and use of fully-automated, robust, intelligent mapping systems** (see further §3.7.3). Systems that will be able to detect possible problems in the data, iteratively estimate the most reasonable model parameters, employ all possible auxiliary and empirical data, and assist the user in generating

the survey reports. Certainly, in the near future, a prediction model will be able to run more analysis and offer more information. This might overload the inexperience users, so that practical guides even thicker than this one can be anticipated.

Important sources:

★ Hengl T., Heuvelink G. B. M., Rossiter D. G., 2007. About regression-kriging: from equations to case studies. Computers and Geosciences, in press.

★ Pebesma, E., 2004. Multivariable geostatistics in S: the gstat package. Computers & Geosciences 30: 683–691.

★ Christensen, R. 2001. Best Linear Unbiased Prediction of Spatial Data: Kriging. In: Cristensen, R. "Advanced Linear Modeling", Springer, 420 pp.

★ Pebesma, E. J., 1999. Gstat user's manual. Department of Physical Geography, Utrecht University, Utrecht, 96 pp.

★ Stein, M. L., 1999. Interpolation of Spatial Data: Some Theory for Kriging. Series in Statistics. Springer, New York, 247 pp.

3

Hands-on software

3.1 Overview and installation of software

This chapter will introduce you to the four packages that we will use to run the processing and display the results: ILWIS, SAGA, R+gstat and Google Earth. All these are available as open source or as freeware and no licenses are needed to use them. By combining the capabilities of the four packages we get the best out of each package and optimize both preparation, processing and the visualization of the maps. In this case, ILWIS GIS will be primarily used to process and prepare vector and raster maps and run simple analysis; SAGA GIS will be used to run analysis on DEMs, but also for geostatistical interpolations; R+gstat will be used for various types of statistical and geostatistical analysis, but also for data processing automation; Google Earth will be used only to visualize the results and prepare the final layouts. Follow the instructions down-below to install these packages and make first steps in them.

3.1.1 ILWIS

ILWIS (Integrated Land and Water Information System) is a stand-alone integrated GIS package developed at the International Institute of Geoinformation Science and Earth Observations (ITC), Enschede, Netherlands. ILWIS was originally built for educational purposes and low-cost applications in developing countries. Its development started in 1984 and the first version (DOS version 1.0) was released in 1988. ILWIS 2.0 for Windows was released at the end of 1996, and a more compact and stable version 3.0 (WIN 95) was released by mid 2001. From 2004, ILWIS was distributed solely by ITC as shareware at a nominal price, and from July 2007, ILWIS shifted to open source. ILWIS is now freely available ('as-is' and free of charge) as open source software (binaries and source code) under the 52°North initiative).

The most recent version of ILWIS (3.4) offers a range of image processing, vector, raster, geostatistical, statistical, database and similar operations (Unit Geo Software Development, 2001). In addition, a user can create new scripts, adjust the operation menus and even build Visual Basic, Delphi, or C++ applications that will run at top of ILWIS and use its internal functions. In principle, the biggest advantage of ILWIS is that it is a compact package with a diverse vector and raster-based GIS functionality and the biggest disadvantages are bugs and instabilities and necessity to import data to ILWIS format from other more popular GIS packages.
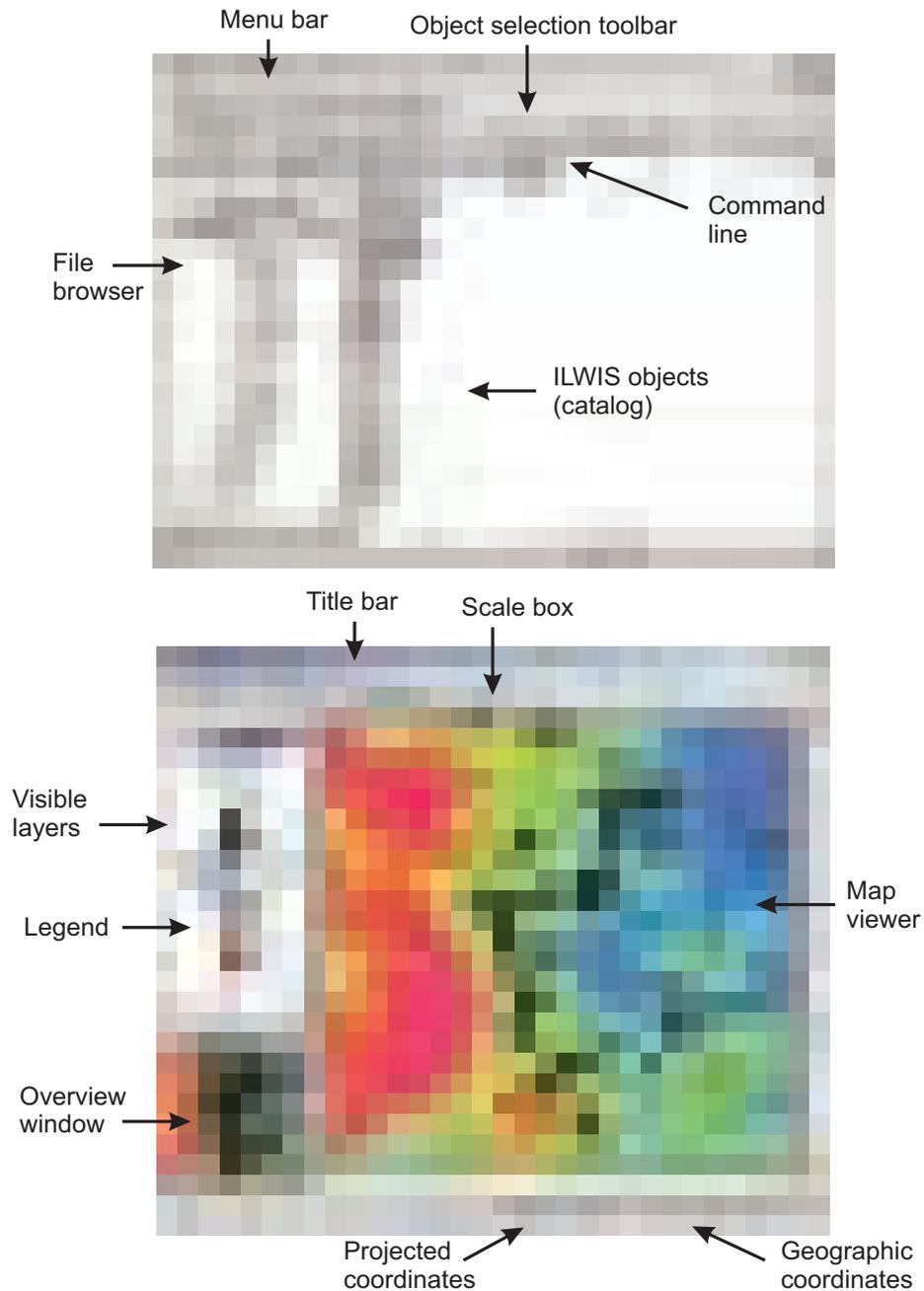
Fig. 3.1: The ILWIS main window (above) and map window (below).

The ILWIS binaries are very simple to install. Download the zipped binaries and unpack them to a local folder e.g. `C:\Program Files\ILWIS\`. In this folder, you will find the `Ilwis30.exe` file, which is the main executable for ILWIS. Double click this file to start ILWIS. You will first see the main program window, which can be compared to the ArcGIS catalog (Fig. 3.1). The main program window is, in fact, a file browser which lists all ILWIS operations, objects and supplementary files within a working directory. The ILWIS Main window consists of a Menu bar, a Standard toolbar, an Object selection toolbar, a Command line, a Catalog, a Status bar and an Operations/Navigator pane with an Operation-tree, an Operation-list and a Navigator. The left pane (Operations/-

Navigator) is used to browse available operations and directories and the right menu shows available spatial objects and supplementary files (Fig. 3.1). GIS layers in different formats will not be visible in the catalog until we define the external file extension.

An advantage of ILWIS is that, every time a user runs an command from the menu bar or operation tree, ILWIS will record the operation in ILWIS command language. For example, you can interpolate a point map using inverse distance interpolation by selecting *Operations ↦ Interpolation ↦ Point interpolation ↦ Moving average*, which will be shown as:

```
ev_idw.mpr = MapMovingAverage(ev, mapgrid.grf, InvDist(1,5000), plane)
```

where `ev_idw.mpr` is the output map, `MapMovingAverage` is the interpolation function, `mapgrid` is the grid definition (georeference) and `InvDist` is the method. This means that you can now edit this command and run it directly from the command line, instead of manually selecting the operations from the menu bar. In addition, you can copy such commands into an ILWIS script to enable automation of data analysis. ILWIS script can use up to nine script parameters, which can be either spatial objects, values or textual strings.

3.1.2 SAGA

SAGA (System for Automated Geoscientific Analyses) is an open source GIS that has been developed since 2001 at the University of Göttingen[1], Germany, with the aim to simplify the implementation of new algorithms for spatial data analysis (Conrad, 2006). It is a full-fledged GIS with support for raster and vector data, which includes a large set of geoscientific algorithms, being especially powerful for the analysis of DEMs. With the release of version 2.0 in 2005, SAGA works under both Windows and Linux operating systems. In addition, SAGA is an open-source package, which makes it especially attractive to users that would like extend or improve its existing functionality.

SAGA handles tables, vector and raster data and natively supports at least one file format for each data type. Currently SAGA provides about 42 free module libraries with 234 modules, most of them published under the GPL. The modules cover geo–statistics, geomorphometric analysis, image processing, cartographic projections, and various tools for vector and raster data manipulation. Modules can be executed directly by using their associated Parameters window. After you have imported all maps to SAGA, you can also save the whole project so that all associated maps and visualisation settings are memorized. The most comprehensive modules in SAGA are connected with hydrologic, morphometric and climatic analysis of DEMs.

To install SAGA, download and unzip the files to a local directory e.g. `C:\Program Files\SAGA2\`. Then run the `saga_gui.exe` and you will get a GUI as shown in Fig. 3.2. In addition to the GUI, a second user front end, the SAGA command line interpreter can be used to execute modules.

3.1.3 R

R is the open source version of the S language for statistical computing. Apparently, the name "R" was selected for two reasons: (1) prestige — "R" is a letter before "S", and (2) coincidence — both of the creators' names start with a letter "R". R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests,

---

[1]The group recently collectively moved to the Institut für Geographie, University of Hamburg.
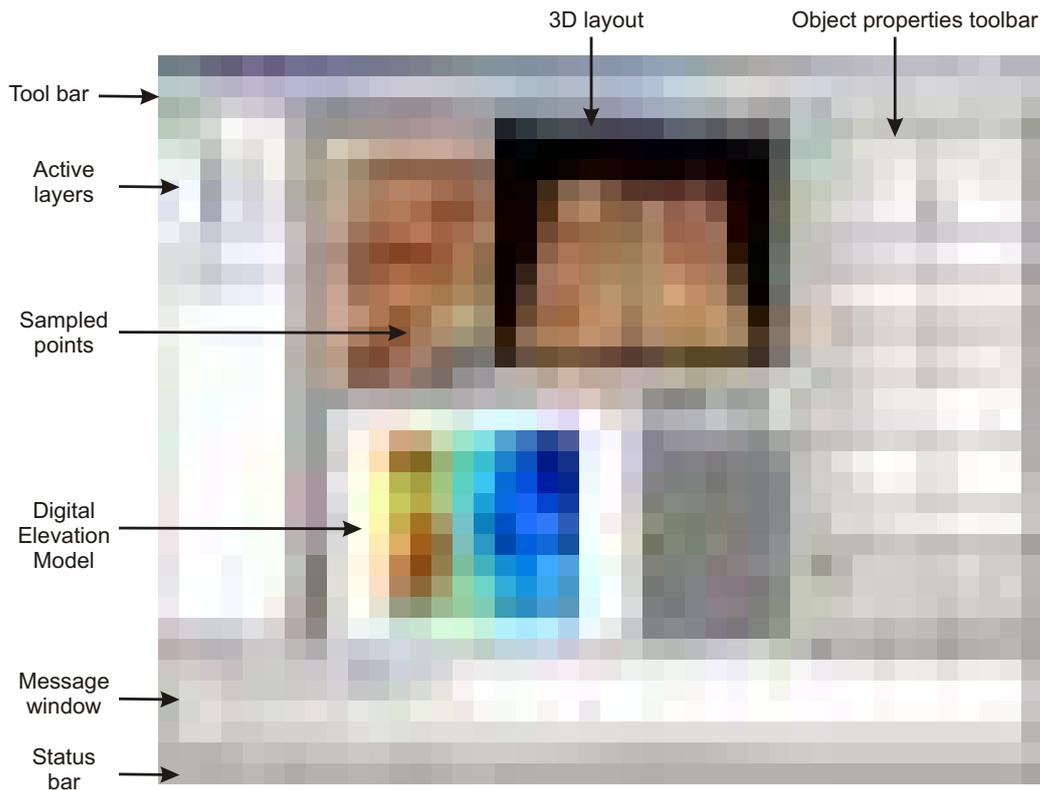
Fig. 3.2: The SAGA GUI elements and displays.

time-series analysis, classification, clustering,. . .) and graphical techniques, and is highly extensible (Chambers and Hastie, 1992; Venables and Ripley, 2002). The S language has often been the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity. It takes a lot of time for the beginners to get use the R syntax, but the benefits are worth the time investment.

To install R under Windows, download and run an installation exe from the R-project homepage. This will install R for window with a GUI. After you start R, you will first need to set-up the working directory and install additional packages. To run geostatistical analysis in R, you will need to add the following R packages: gstat (gtat in R), rgdal (GDAL import of GIS layers in R), sp (operations on maps in R), foreign, spatstat (spatial statistics in R) and maptools.

To install these packages you should do the following. First start the R GUI, then select the *Packages* ↦ *Load package* from the main menu. Note that, if you wish to install a package on the fly, you will need to select a suitable CRAN mirror from where it will download and unpack a package. Another important issue is that, although a package is installed, you will still need to load it into your workspace (every time you start R) before you can use its functionality. A package is commonly loaded using e.g.:

```
> library(gstat)
```

R is today identified as one of the fastest growing and most comprehensive statistical computing tools/communities. It practically offers statistical analysis and visualisation of unlimited sophistication. A user is not restricted to a small set of procedures or options, and because of the contributed packages, users are not limited to one method of
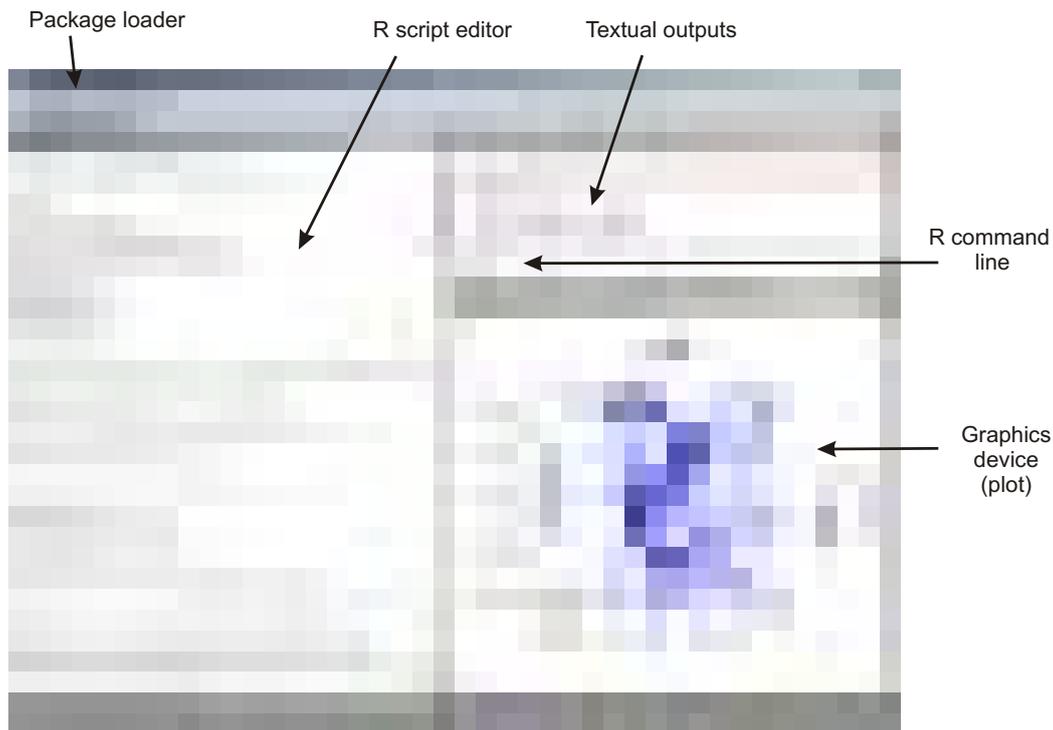
Fig. 3.3: The GUI of R under Windows and typical plot produced using the sp package.

accomplishing a given computation or graphical presentation (Rossiter, 2007a; Murrell, 2006). As we will see later on, R became attractive for geostatistical mapping mainly due to the recent integration of the geostatistical tools (gstat) and tools that allow R computations with spatial data layers (sp, maptools, rgdal).

3.1.4   Gstat

Gstat is a stand-alone package for geostatistical analysis developed by Edzer Pebesma from the University of Utrecht in the Netherlands in 1997. As of 2003, the gstat functionality is also available as an S extension, either as R package or S-Plus library. Current development mainly focuses on the R/S extensions, although the stand alone version can still be used for many applications.

To install gstat (the stand-alone version) under Windows, download the gstat.exe and gstatw.exe (variogram modelling with GUI) files from the www.gstat.org website and put them in your \Windows\system32\ directory. Then, you can always run gstat from the Windows start menu. The gstat.exe runs as a DOS application, which means that there is no GUI. A user controls the processing by editing the command files.

3.1.5   Google Earth

Google Earth is the Google's geographical browser that is increasingly popular in the research community. Google Earth was developed by Keyhole, Inc., a company acquired by Google in 2004. The product was renamed Google Earth in 2005 and is currently

available for use on personal computers running Microsoft Windows 2000 or XP, Mac OS X.

All displays in Google Earth are controlled by the KML files, which are written in the Keyhole Markup Language developed by Keyhole, Inc. KML is an XML-based language for managing the display of three-dimensional geospatial data and is, in fact, used in several geographical browsers (Google Maps, Google Mobile, ArcGIS Explorer and World Wind). The KML file specifies a set of standard features (placemarks, images, polygons, 3D models, textual descriptions, etc.) for display in Google Earth. Each place always has a longitude and a latitude. Other data can make the view more specific, such as tilt, heading, altitude, which together define a *camera view*. The KML datasets can be edited using an ASCII editor (as with HTML), but they can be edited also directly in Google Earth. KML files are very often distributed as KMZ files, which are zipped KML files with a `.kmz` extension.

To install Google Earth, run the `GoogleEarthWin.exe` that you can obtain from the Google's website. To start a KML file, just double-click it and the map will be displayed using the default settings. Other standard Google's background layers, such as roads, borders, places and similar geographic features, can be turned on of off using the Layers panel. There is also a commercial Plus and Pro versions of Google Earth, but for purpose of exercises in this guide, the free version is more than enough.

3.2 Geostatistics in **ILWIS**

ILWIS has a number of built-in statistical and geostatistical functions. Considering the interpolation possibilities, it can be used to analyse prepare a variogram, analyse the anisotropy in the data (including the variogram surface), run ordinary kriging and co-kriging (with one covariable), universal kriging with coordinates[2] as predictors and linear regression. ILWIS has also a number of original geostatistical algorithms that can often be handy for use.

For example, it offers direct kriging from raster that can be used to filter the missing values in a raster map, and a direct calculation of variograms from raster data. ILWIS is also very handy to run some basic statistical analysis on multiple raster layers (map lists): it offers principal component analysis on rasters, correlation analysis between rasters and multi-layer map statistics (min, max, average and standard deviation).

Although ILWIS can not be used to run regression-kriging as defined in §2.1.3, it can be used to run a similar type of analysis. For example, a table can be imported and converted to a

Fig. 3.4: Overlay between points and rasters: table calculation in ILWIS.

point map using the *Table to PointMap* operation. The point map can then be overlaid over raster maps to analyze if the two variables are correlated. This can be done by

---

[2]In ILWIS, the term *Universal kriging* is used exclusively for interpolation of point data without any auxiliary maps.

using a table calculation and `MapValue` function (Fig. 3.4):

```
 TWI=mapvalue(TWI, coord(X,Y,germany3))
```

where `mapvalue` is the overlay function that obtains values of a map `TWI` by using coordinates `X`, `Y` of the points and the `germany3` coordinates system. In the same table, you can then derive a simple linear regression model e.g. `SAND = b0 + b1 * TWI`. By fitting a least square fit using a polynomial, you will get: `b0=67.985` and `b1=-4.429`. This means that the sand content decreases with an increase of `TWI` — Topographic Wetness Index (Fig. 3.5). Note that, in ILWIS, you can not derive the Generalized Least Squares (GLS) regression coefficients (Eq.2.1.3) but only the OLS coefficients, which is statistically suboptimal because the residuals are possibly auto-correlated (see §2.1). In fact, regression modelling in ILWIS is so limited that I can only advise you to always export the table data to R and then run statistical analysis. After you estimate the regression coefficients, you produce a map of SAND content (deterministic part of variation) by running a map calculation:

```
SAND_lm=67.985-4.429*TWI
```

Now you can estimate the residuals at sampled locations using a table calculation as in Fig. 3.4:

```
SAND_res=SAND-MapValue(SAND_lm, coord(X,Y,germany3))
```

You can create a point map for residuals and derive a variogram of residuals by using operations *Statistics ↦ Spatial correlation* from the main menu. If you use a lag spacing of 800 m, you will get a variogram that can be fitted[3] with a spherical variogram model (`C0=180`, `C1=220`, `R=1800`). The residuals can now be interpolated using ordinary kriging, which gives typical kriging pattern. The fitted trend and residuals can then be added back together using:

```
SAND_RK=SAND_lm+SAND_res_OK
```


Fig. 3.5: Correlation plot `TWI` vs `SAND`.

which gives regression-kriging predictions. Note that, in this case, we will produce negative values for `SAND` in the areas where the `TWI` is high (Fig. 3.5). Also note that, because a complete RK algorithm with GLS estimation of regression is not implemented in ILWIS (§2.1.3), we are not able to derive a map of the prediction variance (Eq.2.1.5).

Raster maps from ILWIS can be exported to other packages. You can always use export them to the Arc/Info ASCII (`.ASC`) format. If the georeference in ILWIS has been set as center of the corner pixels, then you might need to manually edit the `*.asc` header[4]. Otherwise, you will not be able to import such maps to ArcGIS (8 or higher) or Idrisi.

---

[3]ILWIS does not support automated variogram fitting.
[4]Simply replace in the header of the file `xllcenter` and `yllcenter` to `xllcorner` and `yllcorner`.

3.2.1 Visualization of uncertainty using whitening

As mentioned previously, an advantage of ILWIS is that it is a script-based package so that various scripts can be developed that allow data processing automation. One such script that will be used later on in the exercise is the script to visualize uncertainty of the map together with the actual predictions — VIS_error. This script produces colour maps that should be associated with the special 2D legends.
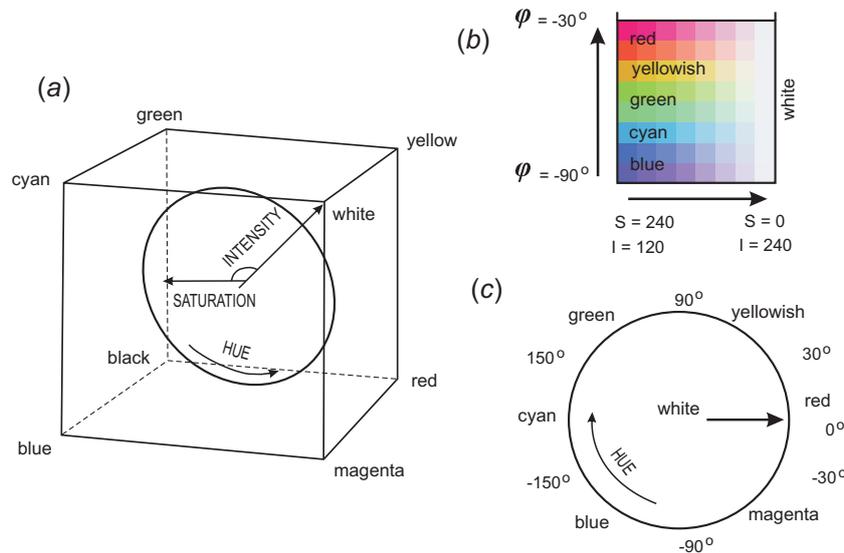

Fig. 3.6: Design of the special 2D legend used to visualize the prediction variance using whitening: (a) the HSI colour model, (b) the 2D legend and (c) the common types of Hues.

This visualization methods is based on the Hue-Saturation-Intensity (HSI) colour model (Fig. 3.6a) and calculations with colours using the colour mixture (CM) concept. The HSI is a psychologically appealing colour model – hue is used to visualise values or taxonomic space and whiteness (paleness) is used to visualise the uncertainty (Dooley and Lavin, 2007). For this purpose, a 2D legend was designed to accompany the visualisations. Unlike standard legends for continuous variables, this legend has two axis (Fig. 3.6b): (1) vertical axis (hues) is used to visualise the predicted values and (2) horizontal axis (whiteness) is used to visualise the prediction error. Fig. 3.7 shows some examples of visualizing the mapping precision for different variables using the same input data. Note how different can be the precision of spatial prediction.

To visualize the uncertainty for your own case study using this technique, you should follow these steps:

(1.) Download the VIS_error script for visualization of prediction error and unzip it to the default directory (C:\Program Files\ILWIS\Scripts\).

(2.) Derive the predictions and prediction variance for some target variable. Import both maps to ILWIS. The prediction variance needs to be then converted to normalized prediction variance by using Eq.(4.6.4), so you will also need to determine the global variance of your target variable.

(3.) Start ILWIS and run the script from the left menu (operations list) or from the

main menu $\mapsto$ *Operations* $\mapsto$ *Scripts* $\mapsto$ VIS_error. Use the help button to find more information about the algorithm.

(4.) To prepare final layouts, you will need to use the legend_hsi.bmp legend file[5].
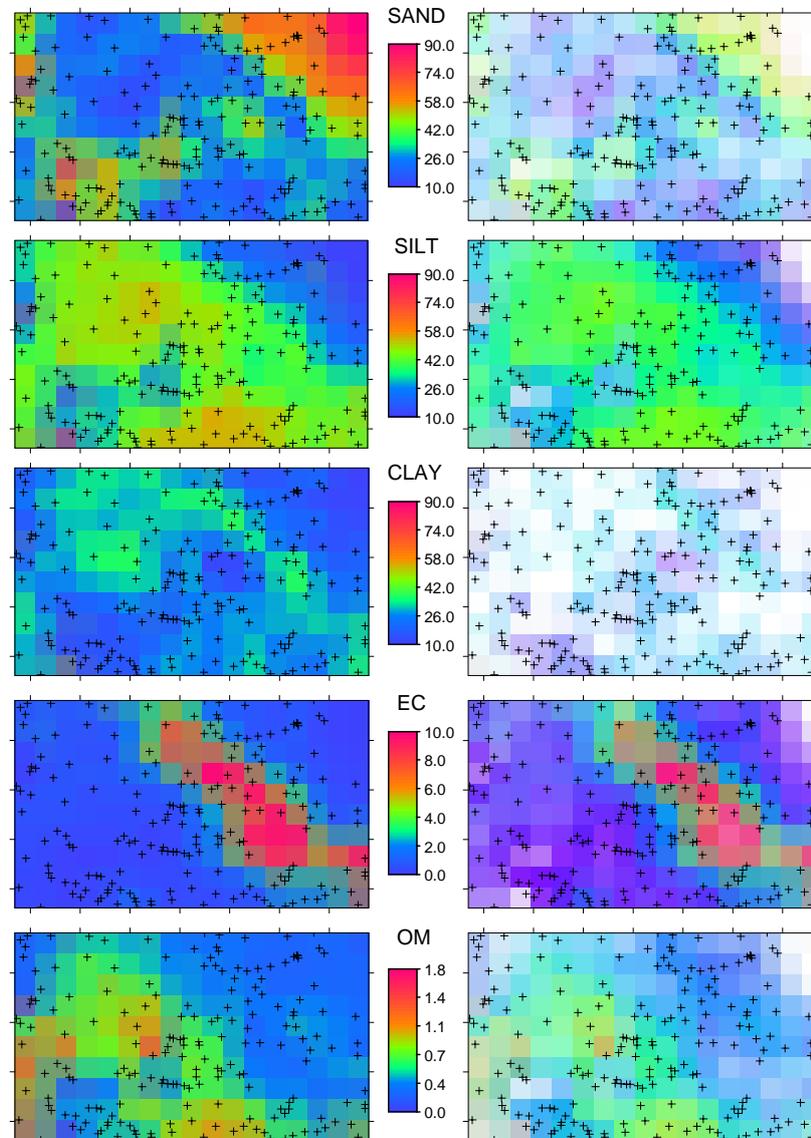

Fig. 3.7: Success of spatial predictions for various soil variables: sand (SAND), silt (SILT) and clay (CLAY) content in %, Electric conductivity (EC) and organic matter (OM) content in %. On the left are the predicted values and on the right are the predictions visualized using whitening. After Hengl and Toomanian (2006).

You can manually change the lower and upper values for both prediction and error maps depending on your mapping requirements. By default, thresholds of 0.4 and 0.8 (max 1.0) are used for the normalized prediction error values. This assumes that a satisfactory prediction is when the model explains more than 85% of the total variation (normalized error = 40%). Otherwise, if the values of the normalized error gets above

---

[5]This legend is a Hue-whitening legend: in the vertical direction only Hue values change, while in the horizontal direction amount of white colour is linearly increased from 0.5 up to 1.0.

80%, the model accounted for less than 50% of variability at the validation points and the prediction is unsatisfactory (Fig. 3.7).

3.3 Geostatistics in **SAGA** GIS

SAGA offers limited capabilities for geostatistical analysis, but in a very user-friendly environment. Note that many commands in SAGA are available only by right-clicking the specific data layers.

For example, you make a correlation plot between two grids by right-clicking a map of interest, then select *Show Scatterplot* and you will receive a module execution window where you can select the second grid (or a shape file) that you would like to correlate with your grid of interest. This will plot all grid-pairs and display the regression model and its significance (R-square) as shown in Fig. 3.8. The setting of the Scatterplot options can be modified by selecting *Scatterplot* from the main menu. Here you can adjust the regression formula, obtain the regression details, and adjust the graphical settings of the scatterplot.

Fig. 3.8: Correlation plot for grids: `TWI` vs `DEM`.

Under the module *Geostatistics*, three groups of operations can be found: (a) *Grid* (various operations on grids); (b) *Points* (derivation of semivariances) and (c) *Kriging* (ordinary and universal kriging). Under the group *Grid*, several modules can be run: *Multiple Regression Analysis* (relates point data with rasters), *Radius of Variance* (detects a minimum radius to reach a particular variance value in a given neighbourhood), *Representativeness* (derives variance in a local neighbourhood), *Residual Analysis* (derives local mean value, local difference from mean, local variance, range and percentile), *Statistics for Grids* (derives mean, range and standard deviation for a list of rasters) and *Zonal Grid Statistics* (derives statistical measures for various zones and based on multiple grids). For the purpose of geostatistical mapping, we are especially interested to correlate points with rasters (§1.3.2), which can be done via the *Multiple Regression Analysis* module. By starting this module you will get a parameter setting window (Fig. 3.9).

By running the *Multiple Regression Analysis* module, SAGA will estimate the values of points at grids, run the regression analysis and predict the values at each location (Fig. 3.10). You will also get a textual output (message window) that will show the regression model, and a list of the predictors according to their importance:

```
Regression:
Y = 84.783626 -4.692293*[TWI] -128.368154*[GSI] -8.050256*[SLOPE]
-0.004243*[DEM]
```

Correlation:
1: R^2 = 14.418404% [14.418404%] -> TWI
2: R^2 = 15.719900% [1.301496%] -> GSI
3: R^2 = 16.595425% [0.875525%] -> SLOPE

```
4:  R² = 16.600508% [0.005082%] -> DEM
```

in this case the most significant predictor is `TWI` and the least significant predictor is `DEM`. The model finally explains 16.6% of the total variation.

Fig. 3.9: Running predictions by using regression analysis in SAGA GIS

.

When selecting the multiple regression analysis options, you can also opt to derive the residuals and fit the variogram of residuals. These will be written as a shape file that can then be used to derive semivariances. Select *Geostatistics ↦ Points ↦ Semivariogram* and specify the distance increment (lag) and maximum distance. The variogram can be displayed by again right clicking a table and selecting *Show Scatterplot* option (Fig. 3.10). At the moment, the regression models in SAGA are limited to linear, exponential and logarithmic models, which is a problem if you need to fit a variogram. You can at least use the logarithmic model which will estimate something close to the exponential variogram model (Eq.1.3.8).
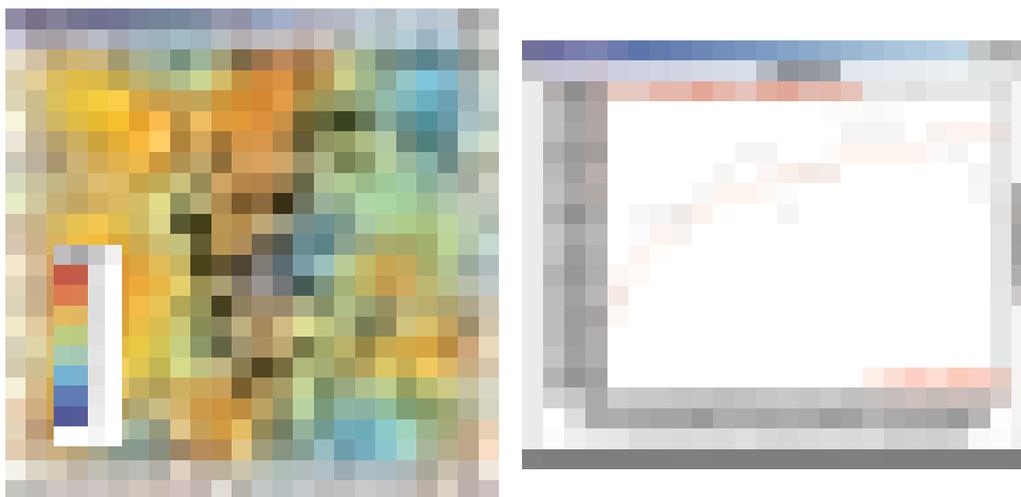
Fig. 3.10: Output of mapping SAND content using `TWI`, `SLOPE`, `GSI` and `DEM` (left) and the resulting variogram of residuals (right).

Once the regression model and the variogram of the residuals have been estimated, a user can also run regression-kriging, which is available in SAGA under the module

Geostatistics $\mapsto$ *Universal kriging.* The users can select global and local (search radius) version of the Universal kriging. In this case, I do not recommend the use of local Universal kriging with a small search radius ($\ll$100 points) because it over-simplifies the technique and can lead to artefacts[6]. Note also that, in SAGA, you can select as many predictors as you wish, as long as they are all in the same grid system. The final results can be visualized in both 2D and 3D spaces.

Another advantage of SAGA is the ability to use script files for the automation of complex work-flows, which can then be applied to different data projects. Scripting of SAGA modules is now possible in two ways:

(1.) Using the command line interpreter (`saga_cmd.exe`) with DOS batch scripts. Some instructions on how to generate batch files can be found in Conrad (2006).

(2.) A much more flexible way of scripting provides the Python interface to the SAGA Application Programming Interface (SAGA-API).

In addition to scripting possibilities, SAGA allows you to save SAGA parameter files (`*.sprm`) that contain all inputs and output parameters set using the module execution window. These parameter files can be edited in an ASCII editor, which can be quite handy to automate processing.

In summary, SAGA GIS has many attractive features for geostatistical mapping: (1) it has a large library of modules, especially to parameterize geomorphometric features of a terrain, (2) it can generate maps from points and rasters by using multiple linear regression and regression-kriging, and (3) it is an open source GIS with a popular GUI. If compared to gstat, SAGA is not able to run geostatistical simulations, GLS estimation nor stratified or co-kriging, however, it is capable of running regression-kriging in a statistically sound way (unlike in ILWIS that can only run OLS predictions). The advantage of SAGA over R is that it can load and calculate with relatively large maps (not recommended in R for example) and that it can be used to visualize the input and output maps in 2D and 3D (map drapes). To get additional support, visit the official website or obtain the SAGA users' manual (Olaya, 2004).

3.4 Geostatistics with **gstat**

Gstat is possibly the most complete and certainly the most accessible geostatistical package in the World. It can be used to calculate sample variograms, fit valid models, plot variograms, calculate (pseudo) cross variograms, and calculate and fit directional variograms and variogram models (anisotropy coefficients are not fitted automatically). Kriging and (sequential) conditional simulation can be done under (simplifications of) the universal co-kriging model. Any number of variables may be spatially cross-correlated. Each variable may have its own number of trend functions specified (being coordinates, or so-called external drift variables). Simplifications of this model include ordinary and simple kriging, ordinary or simple co-kriging, universal kriging, external drift kriging, Gaussian conditional or unconditional simulation or cosimulation. In addition, variables may share trend coefficients (e.g. for collocated co-kriging). To learn about capabilities of gstat, a user is advised to read the gstat User's manual, which is still by far the most complete documentation about gstat package.

As mentioned previously, gstat can be run as a stand-alone application, or as a R package. Gstat is also implemented in the most recent version of Idrisi GIS with a GIS

---

[6]Neither local variograms nor local regression models are estimated. See §2.2 for a detailed discussion.

GUI. Although gstat is currently maintained as a S/R package primarily, it might be advisable to compare different software options for real mapping projects.

3.4.1 The stand-alone version of **gstat**

In the stand-alone version of the gstat, everything is done via compact scripts or command files. The best approach to prepare the command files is to learn from the list of example command files that can be found in the gstat User's manual. Preparing the command files for gstat is rather simple and fast. For example, to run inverse distance interpolation the command file would look like this:

```
# Inverse distance interpolation on a mask map
data(ev):  'points.eas', x=1, y=2, v=3;
mask:  'mapgrid.asc'; # the prediction locations
predictions(ev):  'ev_idw.asc'; # result map
```

where the first line defines the input point dataset (points.eas[7]), the coordinate columns (x, y) are the first and the second column in this table, and the variable of interest is in the third column; the prediction locations are the grid nodes of the map mapgrid.asc[8] and the results of interpolation will be written to a raster map ev_idw.asc.

To extend the predictions to regression-kriging, the command file needs to include the auxiliary maps and the variogram model for the residuals:

```
# Regression-kriging using two auxiliary maps
data(ev):  'points.eas', x=1, y=2, v=3, X=4,5;
variogram(ev):  1 Nug(0) + 5 Exp(1000);
mask:  'q1.asc', 'q2.asc'; # the predictors
predictions(ev):  'ev_idw.asc'; # result map
```

where X defines the auxiliary predictors, 1 Nug(0) + 5 Exp(1000) is the variogram of residuals and q1.asc and q2.asc are the auxiliary predictors. All auxiliary maps need to have the same grid definition and need to be available also in the input table. The program will also fail if the predictors are not sorted in the same order in both first and the third line. Note that there are many optional arguments that can be included in the command file: a search radius can be set using "max=50"; switching from predictions to simulations can be done using "method: gs"; bloc kriging can be initiated using "blocksize: dx=100" etc.

To run a command file start DOS prompt by typing: > cmd, then move to the active directory by typing: e.g. > cd c:\gstat; to run spatial predictions or simulations run the gstat programme together with a specific gstat command file from the DOS prompt (Fig. 3.11):

```
> gstat.exe ev_rk.cmd
```

Gstat can also automatically fit a variogram by using:

---

[7]An input table in the GeoEAS format.
[8]Typically ArcInfo ASCII format for raster maps.
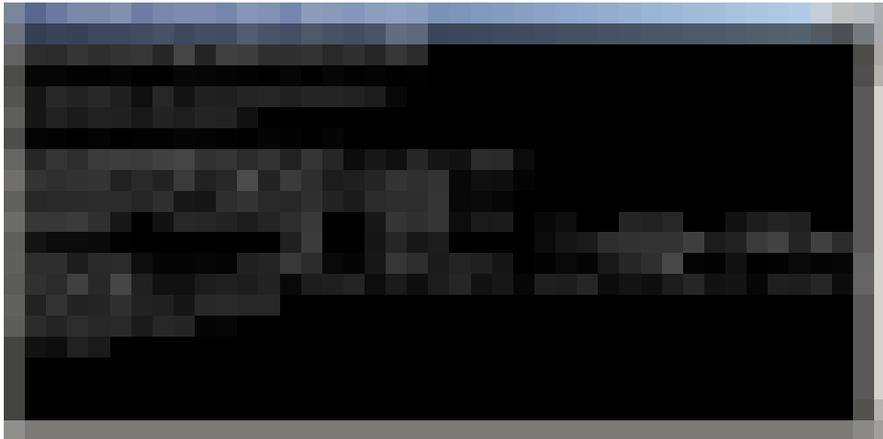
Fig. 3.11: Running interpolation using the gstat stand-alone: the DOS command prompt.

```
data(ev):  'points.eas', x=1,y=2,v=3;

# set an initial variogram:

variogram(ev):  1 Nug(0) + 5 Exp(1000);

# fit the variogram using standard weights:

method:  semivariogram;
set fit=7;
# write down the fitted variogram model and gnuplot

set output= 'vgm_ev.cmd';
set plotfile= 'vgm_ev.plt';
```

where set `fit=7` defines the fitting method (weights=$N_j/\mathbf{h}_j^2$), `vgm_ev.cmd` is the text file where the fitted parameters will be written. Once you fitted a variogram, you can then view it using the wgnuplot application). Note that, for automated modelling of variogram, you will need to define the fitting method and an initial variogram, which is then iteratively fitted against the sampled values. Edzer Pebesma suggest use of initial exponential variogram with nugget parameter = measurement error, sill parameter = sampled variance, and range parameter = 10% of the spatial extent of the data (or two times the mean distance to the nearest neighbour). This can be termed a **standard initial variogram model**. Although the true variogram can be quite different, it is important to have a good idea of how the variogram *should* look like.

There are many advantages of using the stand-alone version of gstat. The biggest ones are that it takes little time to prepare a script and that it can work with large maps (unlike R that often faces vector allocation problems). In addition, the results of interpolation are directly saved in a GIS format and can be loaded to ILWIS or SAGA. However, for regression-kriging, we need to estimate the regression model first, then derive the residuals and estimate their variogram model, which can not be automated in gstat so we anyway need to load the data to some statistical package before we can prepare the command file. Hence, the stand-alone version of gstat, as SAGA, can be used for geostatistical mapping, but only once all the regression-kriging parameters have been estimated.

3.4.2 Geostatistics in R

Although there are several packages in R to do geostatistical analysis and mapping, many recognize R+gstat as the only complete and fully-operational package, especially if you wish to run regression-kriging, multivariate analysis, geostatistical simulations and block predictions (Hengl et al., 2007b; Rossiter, 2007b). To allow extension of R functionalities to operations with spatial data, the developer of gstat, with a support of colleagues, have develop a R package called sp (Pebesma and Bivand, 2005). Now the users are able to load GIS layers directly into R, run geostatistical analysis on grid and points and display spatial layers as in a standard GIS package. In addition to sp, two important spatial data protocols have also been recently integrated in R: (1) **GIS data exchange protocols** (GDAL — Geospatial Data Abstraction Library, and OGR — OpenGIS Simple Features Reference Implementation), and (2) **map projection protocols** (PROJ.4 — Cartographic Projections Library). This allows R users to import/export raster and vector maps, run raster/vector based operations and combine them with statistical computing functionality of various packages. The development of GIS and graphical functionalities within R has already caused a small revolution and many GIS analysts are seriously thinking about completely shifting to R (Pebesma and Bivand, 2005; Murrell, 2006).

A complete overview of gstat functions and example of R commands is given in Pebesma (2004)[9]. The most used gstat functions in R include:

- `variogram` — calculates sample (experimental) variograms;

- `plot.variogram` — plots an experimental variogram with automatic detection of lag spacing and maximum distance;

- `fit.variogram` — iteratively fits an experimental variogram using reweighted least squares estimation;

- `krige` — a generic function to make predictions by inverse distance interpolation, ordinary kriging, OLS regression, regression-kriging and co-kriging;

- `krige.cv` — runs krige with cross-validation using the *n*-fold or *leave-one-out* method;

Typically, a function in R (called method) consists of four elements — function name, required arguments, optional arguments and output object name (see also page 30):


```
ev.id = krige(ev~1, data=points, newdata=mapgrid, nmax=60)
```
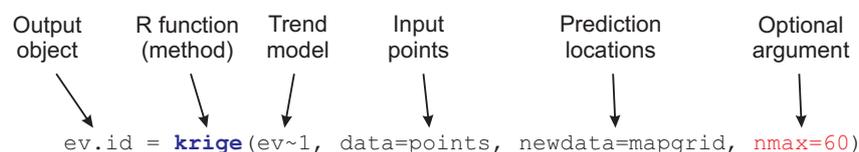
R offers much more flexibility than the stand-alone version of gstat, because users can extend the optional arguments and combine them with outputs or functions derived from other R packages. For example, instead of using a trend model with a constant (intercept), one could use outputs of a linear model fitting, which allows even more compact scripting.

---

[9]The Pebesma (2004) paper reached the TOP25 articles within the subject area Earth and Planetary Sciences on science direct — find out why!

Note that in R, the user must type commands to enter data, do analyses, and plot graphs. If a single argument in the command is incorrect, inappropriate or mistyped, you will get an error message. If the error message is not helpful, then try receiving more help about some operation by typing e.g. `help(krige)` or `?krige` commands or via the Html help files. R is also supported by comprehensive technical documentation and user-contributed tutorials. Visit the R website and look under Documentation section for basic manuals in R. Many very useful introductory notes and books, including translations of manuals into other languages than English, are available from the Contributed documentation section. Another very useful source of information is the R News[10] newsletter, which often offers many practical examples of data processing. You may also wish to register to the special interest groups such as R-sig-Geo or similar and subscribe to their mailing lists. Gstat-info is the mailing list of the gstat package and usually offers many interesting information.

Unfortunately, the help documentation for regression-kriging in R is limited to few code examples only, often without any explanation what it does and how can it be used for mapping of environmental variables. One of the main motives to produce this guide was to diminish this gap.

3.5   Visualisation of maps in **Google Earth**

The rapid emergence and uptake of Google Earth may be considered evidence for a trend towards a more visual approach to spatial data handling. Google Earth's sophisticated spatial indexing of very large datasets combined with an open architecture for integrating and customising new data is having a radical effect on many Geographic Information Systems (Wood, 2007). One of its biggest impacts is that it has opened up the exploration of spatial data to a much wider non-expert community of users (see further §4.8). Google Earth is a winning software in at least five categories:

Availability — It is a free browser that is available to everyone. Likewise, users can upload their own geographic data and share it with anybody (or with selected users only).

High quality background maps — The background maps (remote sensing images, roads, administrative units, topography) are constantly updated and improved. At the moment, almost 20-30% of the World coverage is available in high resolution (2 m IKONOS images). All these layers have been georeferenced at relatively high quality and can always be used to validate spatial accuracy of maps you produce.

A single coordinate system — The geographic data in Google Earth is visualized using a 3D model (central projection) rather than a projected 2D system. This practically eliminates all the headaches you had with understanding projection systems and merging maps from different projection systems. However, always have in mind that the printed Google Earth displays although they might appear to be 2D, will always show distortions due to Earth's curvature (or due to the relief displacements). At very detailed scales (blocks of the buildings), these distortions can be ignored so that the distances on the screen correspond closely to the distances in the nature.

---

[10]Vol. 1/2 of R News, for example, is completely dedicated to spatial statistics in R; see also Pebesma and Bivand (2005) for an overview of classes and methods for spatial data in R.

Web-based data sharing — Google Earth data is located on internet servers so that the users do not need to download or install any data locally.

Popular interface — Google Earth, as many other Google's product, are completely user-oriented. What makes Google Earth especially popular is the impression of literarily flying over Earth's surface and interactively exploring the content of various spatial layers.

There are several competitors to Google Earth (NASA World Wind, ArcGIS Explorer, 3D Weather Globe), although none of them can be compared to Google Earth in any of the above-listed aspects. On the other hand, Google Earth poses some copyright limitations, so you should definitely read the Terms and Conditions before you decide to use it for your own projects. For example, Google welcomes you to use any of the multimedia produced using Google tools as long as you preserve the copyrights and attributions including the Google logo attribution. However, you cannot sell these to others, provide them as part of a service, or use them in a commercial product such as a book or TV show without first getting a rights clearance from Google.

While at present, Google Earth is primarily used as a geo-browser for exploring spatially referenced data, its functionality can be integrated with the geostatistical tools and stimulate sharing of environmental data between international agencies and research groups (Wood, 2007). Although Google Earth does not really offers much GIS functionality, it can be used also to add content, such as points or lines to the existing maps, measure areas and distances, derive UTM coordinates and eventually load GPS data. Still, the biggest use of Google Earth are its visualisation capabilities that can not be compared to any GIS. The base maps in Google Earth are extensive and of high quality, both considering the spatial accuracy and content. In that sense, Google Earth is a GIS that exceeds any existing public GIS in the world.

To load your own GIS data to Google Earth, there are few possibilities. First, you need to understand that there is a difference between loading the vector and raster maps to Google Earth. Typically, it is relatively easy to load vector data such as points or lines to Google Earth, and somewhat more complicated to do the same with raster maps. Also note that, because Google Earth works exclusively only with Latitude/Longitude projection system (WGS84 ellipsoid), all vector/raster maps need to be first reprojected before they can be exported to KML format. More about importing the data to Google Earth can be found via the Google Earth User Guide.

3.5.1 Exporting vector maps to KML

Vector maps can be loaded by using various plugins/scripts in packages such as ArcView, MapWindow and R. Shape files can be directly transported to KML format by using the ArcView's SHAPE 2 KML script, courtesy of Domenico Ciavarella. To install this script, download it, unzip it and copy the two files to your ArcView 3.2 program directory:

- `..\ARCVIEW\EXT32\shape2KML.avx`

- `..\ARCVIEW\network\shp2kmlSource.apr`

This will install an extension that can be easily started from the main program menu (Fig. 3.12). Now you can open a layer that you wish to convert to KML and then click on the button to enter some additional parameters. There is also a commercial plugin for ArcGIS called Arc2Earth, which allows various export options.

Fig. 3.12: Exporting ESRI shape files to KML using an ESRI script in ArcView 3.2. Note that the vector maps need to be first reprojected to LatLon WGS84 system.

An alternative to export the shape files to KML is the Shape2Earth plugin for the open-source GIS MapWindow. Although MapWindow is an open-source GIS, the Shape2Earth plugin is a shareware so you might need to purchase it. Export to KML is possible also in R via the `writeOGR` command. This can be achieved in three steps:

```
# load the rgdal package for GIS data exchange:
require(rgdal)
# reproject the original map to the longlat system:
points.longlat = spTransform(points, CRS("+proj=longlat"))
# export the point map using the KML driver:
writeOGR(points.longlat, "points.kml", "Ebergotzen", driver = "KML")
```

3.5.2 Exporting raster maps (images) to KML

Rasters can not be exported that easy to KML. Google Earth does not allow import of GIS raster formats, but only input of images that can then be draped over a terrain (**ground overlay**). The images need to be exclusively in one of the following formats: JPG, BMP, GIF, TIFF, TGA and PNG. Typically, export of raster maps to KML follows these steps:

(1.) Determine the grid system of the map in the `LatLonWGS84` system. You need to determine five parameters: southern edge (`south`), northern edge (`north`), western edge (`west`), eastern edge (`east`) and `cellsize` in arcdegrees (Fig. 3.13).

(2.) Reproject the original raster map using the new `LatLonWGS84` grid.

(3.) Export the raster map using a graphical format (e.g. TIFF), and optionally the corresponding legend.

(4.) Prepare a KML file that includes a JPG of the map (Ground Overlay), legend of the map (Screen Overlay) and description of how the map was produced. The JPG images you can locate on some server and then refer to an URL.

The bounding coordinates and the cell size in the `LatLonWGS84` coordinate system can be estimated by using the following table calculation in ILWIS. First determine the minimum and maximum corners of your map and convert them to the `LatLonWGS84` system:

```
minX=mincrdx(SAND_rk)
minY=mincrdy(SAND_rk)
maxX=maxcrdx(SAND_rk)
maxY=maxcrdy(SAND_rk)


west1{dom=Value, vr=-180.00000:180.00000:0.00001}=crdx(transform(coord(minX,
minY, germany3), LatlonWGS84))
west2{dom=Value, vr=-180.00000:180.00000:0.00001}=crdx(transform(coord(minX,
maxY, germany3), LatlonWGS84))
east1{dom=Value, vr=-180.00000:180.00000:0.00001}=crdx(transform(coord(maxX,
minY, germany3), LatlonWGS84))
east2{dom=Value, vr=-180.00000:180.00000:0.00001}=crdx(transform(coord(maxX,
maxY, germany3), LatlonWGS84))
```

```
south1{dom=Value, vr=-90.00000:90.00000:0.00001}=crdy(transform(coord(minX,
minY, germany3), LatlonWGS84))
south2{dom=Value, vr=-90.00000:90.00000:0.00001}=crdy(transform(coord(maxX,
minY, germany3), LatlonWGS84))
north1{dom=Value, vr=-90.00000:90.00000:0.00001}=crdy(transform(coord(minX,
maxY, germany3), LatlonWGS84))
north2{dom=Value, vr=-90.00000:90.00000:0.00001}=crdy(transform(coord(maxX,
maxY, germany3), LatlonWGS84))

west=min(west1,west2)
east=max(east1,east2)
south=min(south1,south2)
north=max(north1,north2)
latm=abs((south+north)/2)
```

The cell size and `nrows` and `ncols` for the new map can be determined using
(Fig. 3.13):

```
gpix{dom=Value, vr=0.00000:20.00000:0.00001}=abs(pixsize(%1)/
   (111319*cos(degrad(latm))))
nrows{dom=Value, vr=0:10000:1}=(north-south)/gpix
ncols{dom=Value, vr=0:10000:1}=(east-west)/gpix
```

You can also use an ILWIS script to automatically estimate the bounding coordinates
and produce a new grid definition that matches the borders of your area. Once you have
resampled the map, then export it as an image and copy to some server. A KML file
that can be used to visualize a result of geostatistical mapping would look like this:
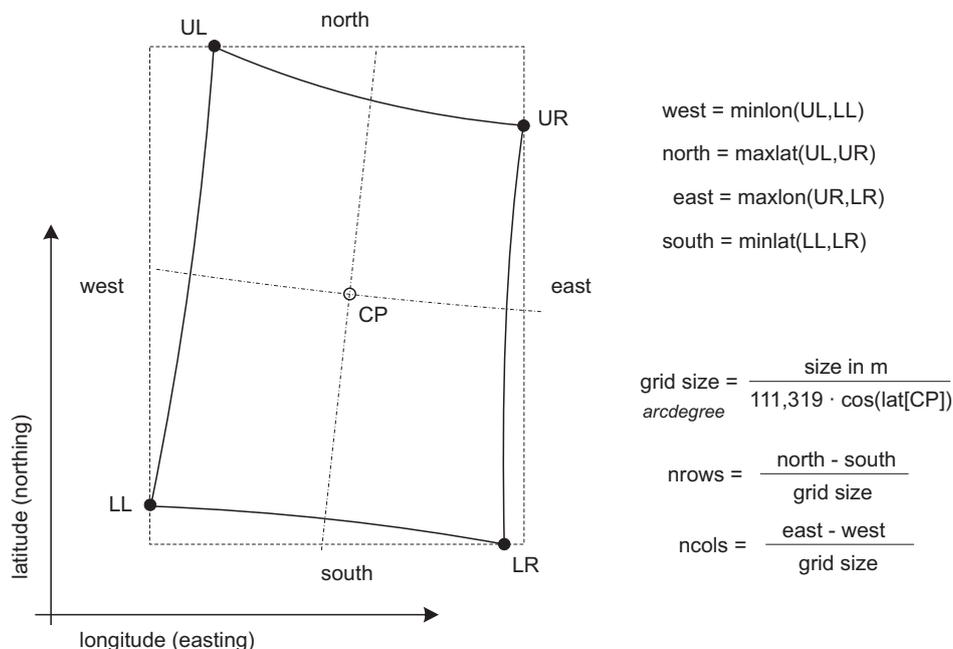
Fig. 3.13: Determination of the bounding coordinates and cell size in the `LatLonWGS84` geographic
projection system using an existing Cartesian system. For large areas (continens), it is advisable
to visually validate the estimated values.

Fig. 3.14: Preparation of the image ground overlays using the Google Earth menu.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
<Document>
    <name>Raster map example</name>
    <GroundOverlay>
        <name>Map name</name>
        <description>Description of how was map produced.</description>
        <Icon>
                <href>http://pedometrics.org/geostat/sand_rk.jpg</href>
            </Icon>
            <LatLonBox>
                <north>51.591667</north>
                <south>51.504167</south>
                <east>10.151389</east>
                <west>10.010972</west>
            </LatLonBox>
    </GroundOverlay>
    <ScreenOverlay>
        <name>Legend</name>
        <Icon>
            <href>http://pedometrics.org/geostat/sand_rk_legend.jpg</href>
        </Icon>
        <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
        <screenXY x="0" y="1" xunits="fraction" yunits="fraction"/>
        <rotationXY x="0" y="0" xunits="fraction" yunits="fraction"/>
        <size x="0" y="0" xunits="fraction" yunits="fraction"/>
    </ScreenOverlay>
</Document>
</kml>
```

In this case the output map (`sand_rk.jpg`) and the associated legend are both placed directly on a server. The resulting map can be seen further in the §4.8. Once you open this map in Google Earth, you can edit it, modify the transparency, change the icons used and combine it with other vector layers. Ground Overlays can also be added directly in Google Earth by using commands *Add ↦ Image Overlay*, then enter the

correct bounding coordinates and location of the image file (Fig. 3.14). Because the image is located on some server, it can also be automatically refreshed and/or linked to a Web Mapping Service (WMS). For a more sophisticated use of Google interfaces see the Mike Williams' Google Maps API tutorial.

3.6   Other software options

3.6.1   Isatis

Isatis[11] is probably the most expensive geostatistical package (>10K €) available in the market today, but is definitively also one of the most professional packages for environmental sciences. Isatis was originally built for Unix, but there are MS Windows and Linux versions also. From the launch of the package in 1993, >1000 licences have been purchased worldwide. Standard Isatis clients are Oil and Gas companies, consultancy teams, mining corporations and environmental agencies.

Isatis offers a wide range of geostatistical functions ranging from 2D/3D isotropic and directional variogram modelling, univariate and multivariate kriging, punctual and block estimation, drift estimation, universal kriging, collocated co-kriging, kriging with external drift, kriging with inequalities (introduce localized constraints to bound the model), factorial kriging, disjunctive kriging etc. From all these, especially, interactivity of exploratory analysis, variogram modelling, detection of local outliers and anisotropy is brought in Isatis to perfection (Fig. 3.15).
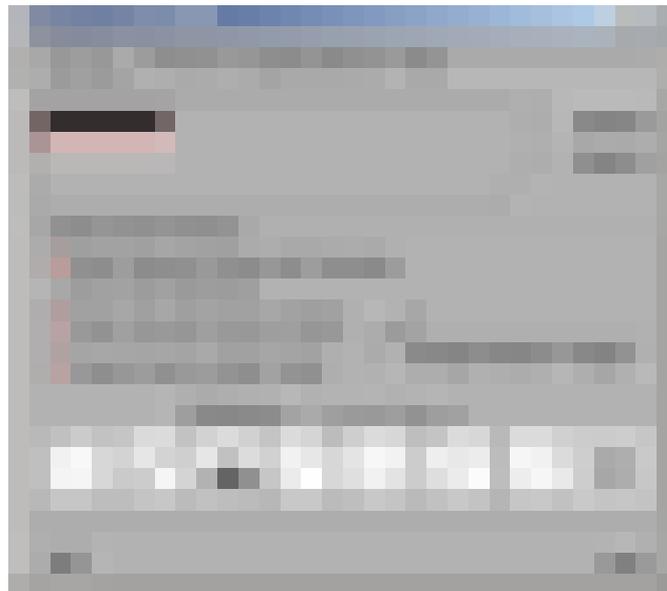
Fig. 3.15: Exploratory data analysis possibilities in Isatis.

Regression-kriging in Isatis can be run by selecting *Interpolation* ↦ *Estimation* ↦ *External Drift (Co)-kriging* (Fig. 3.16). Here you will need to select the target variable (point map), predictors and the variogram model for residuals. You can import the point and raster maps as shape files and raster maps as ArcView ASCII grids (importing/exporting options are limited to standard GIS formats). Note that, before you can do any analysis, you first need to define the project name and working directory using

---

[11]The name is not an abbreviation. Apparently, the creators of Isatis were passionate climbers so they name their package after one climbing site in France.

the data file manager. After you imported the two maps, you can visualize them using the display launcher.

Note that KED in Isatis is limited to only one (three when scripts are used) auxiliary raster map (called *background variable* in Isatis). Isatis justifies limitation of number of auxiliary predictors by computational efficiency. In any case, a user can first run factor analysis on multiple predictors and then select the most significant component, or simply use the regression estimates as the auxiliary map.

Fig. 3.16: Setting the options for Kriging with External Drift in Isatis.

Isatis offers a variety of options for the automated fitting of variograms. You can also edit the *Model Parameter File* where the characteristics of the model you wish to apply for kriging are stored. The user manual does not give insight into the algorithm used (except for the mathematical proofs), so I could not tell how exactly are the KED weights estimated and is there any regression modelling involved prior to the interpolation of data.

3.6.2 GRASS GIS

GRASS (Geographic Resources Analysis Support System) is a general-purpose Geographic Information System (GIS) for the management, processing, analysis, modelling and visualisation of many types of georeferenced data. It is Open Source software released under GNU General Public License. The main component of the development and software maintenance is built on top of highly automated web-based infrastructure sponsored by ITC-irst (Centre for Scientific and Technological Research) in Trento, Italy with numerous worldwide mirror sites. GRASS includes functions to process raster maps, including derivation of descriptive statistics for maps, histograms, but also to generate statistics for time series. There are also several unique interpolation techniques. One should definitively consider using the Regularized spline with tension (RST) interpolation, which has been quoted as one of the most sophisticated methods to generate

smooth surfaces from point data (Mitasova et al., 2005).

The geostatistical functionality in GRASS is achieved mainly via a link to R, actually through an R package called spgrass6 (Bivand, 2005). In the version v5.0 of GRASS, several basic geostatistical functionalities existed including ordinary kriging and variogram plotting, however, the developer of GRASS finally concluded that there is no need to build geostatistical functionality from scratch when a complete open source package already exist. The current philosophy focuses on making GRASS functions also available in R, so that both GIS and statistical operations can be integrated in a single command line. A complete overview of the Geostatistics and spatial data analysis functionality can be found via the GRASS website. Certainly, if you are an Linux user and already familiar with GRASS, you will probably not have many problems to implement the procedures described in chapter §4 and currently adjusted to ILWIS/SAGA GIS.

3.6.3 Idrisi

Idrisi is one of the medium-cost GIS packages but possibly with largest numbers of raster and vector operations. The price per a single licence is about €500, but you can always order a 7-day evaluation version to test it. Idrisi provides statistical tools for spatial analysis of raster images, including simple regression, autocorrelation analysis, pattern analysis, trend analysis, logistical regression, and many more. In Idrisi, gstat code has been also adjusted and integrated within the GUI, which practically means that you can use Idrisi as a graphical user interface to gstat. Additional integration of statistical and GIS functionality can be achieved through a link to S-Plus.
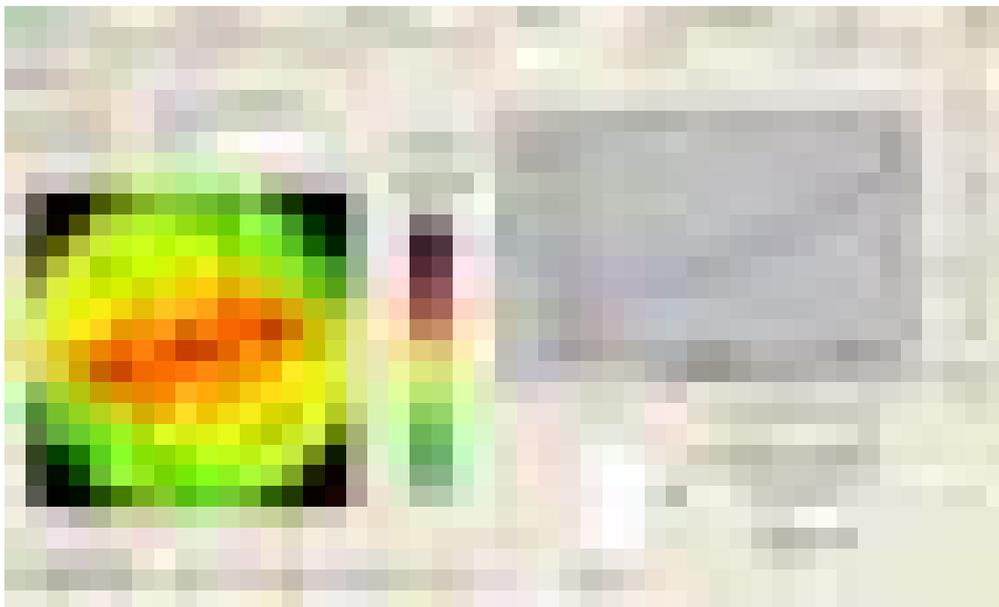
Fig. 3.17: The spatial dependence modeller in Idrisi.

To run regression-kriging in Idrisi, first import all point and raster maps using *Import ↦ Software-specific-formats ↦ ESRI formats ↦ ARCRASTER* (Arcinfo ASCII format to raster); or *SHAPEIDR* (Shape file to Idrisi. First run multiple regression analysis using the points and rasters and then derive the residuals (at sampling locations). You can now model a variogram of residuals by using the Spatial dependence modeller, which also allows modelling of the anisotropy (Fig. 3.17).

Derive the semivariances using a point map, save them and load them in the variogram model fitting environment, where you can (interactively) estimate the variogram model parameters. After you finished fitting the variogram, you can save the variogram parameters and load them later to do kriging or simulations. Once you fitted the variogram model, you can run regression-kriging by selecting *GIS analysis $\mapsto$ Surface analysis $\mapsto$ Interpolation $\mapsto$ Kriging and simulations*. You will then get an user-friendly kriging dialog window (Fig. 3.18) where you can input: variogram model, variable to predict, auxiliary maps and several other optional arguments.

Fig. 3.18: Universal kriging input window in Idrisi.

In Idrisi, only raster input files are allowed for input maps. The first input file contains the sample data, the second, third etc input file is a fully sampled surface that has attribute values for all prediction locations. This means that, before running regression-kriging in Idrisi, you need to convert the point map (`*.vct`) to a raster image using the *RASTERVECTOR* operation. Note that a local neighborhood must be specified for each input file. Unfortunately, Idrisi help warns that the Spatial Dependence Modeler can not calculate generalized least squares residuals, although this was originally implemented in gstat. Also note that Idrisi uses term *residuals* with somewhat different meaning — unlike in R, Idrisi will not derive residuals for you but you will have to do it yourself.

3.7   Summary points

3.7.1   Strengths and limitations of geostatistical software

A full integration of GIS and geostatistics has still many obstacles to tackle. At the moment, a large gap exists between what is possible for some (researchers) and what is available to many (users). No GIS package includes all of generalized linear models, variogram fitting, iterative estimation of residual variograms, and kriging, let alone their seamless integration. Likewise, no statistical package can compete with pure GIS packages specialized to process, edit and display vector and/or raster maps.

A comparison of different aspects of geostatistical packages listed by the AI-Geostats group and several well-known GIS packages can be seen in Table 3.1. Although the universal kriging (using coordinates) is available in most geostatistical packages, kriging with external drift with multiple auxiliary maps can be run in only a limited number of packages. From all software in the world, only Isatis, SAGA, and gstat (as stand-alone application or integrated into R, GRASS or Idrisi) offer a possibility to interpolate a variable using (multiple) auxiliary maps (Hengl et al., 2007b). We have tested regression-kriging in all these packages to discover that RK in Isatis is limited to a use of a single (three in script mode) auxiliary maps (Bleines et al., 2004). In Idrisi, GLS regression coefficients can not be estimated and the system is rather unstable. In gstat, both RK predictions and simulations (predictors as base maps) at both point and block support can be run by defining short scripts, which can help automatize interpolation of large amounts of data. However, gstat implements the algorithm with extended matrix (KED), which means that both the values of predictors and of target variable are used to estimate the values at each new location, which for large datasets can by time-consuming or can lead to computational problems (Leopold et al., 2005).

Setting RK in gstat or SAGA to a smaller window search can lead to termination of the program due to the singular matrix problems. In fact, local RK with a global variogram model is not valid because the regression model will differ locally, hence the algorithm should also estimate the variogram model for residuals for each local neighbourhood (as mentioned previously in §2.2). The singular matrix problem will happen especially when indicator variables are used as predictors or if the two predictor maps are highly correlated. Another issue is the computational effort. Interpolation of $\gg 10^3$ points over 1M of pixels can last up to several hours on a standard PC. To run simulations in R+gstat with the same settings will take even more time. This clearly proves that, although KED procedure is mathematically elegant, it might be more effective for real-life applications to fit the trend and residuals separately (the regression-kriging approach) instead of through use of an extended matrix (the KED approach). A limitation of `gstat.exe` is that it is a stand-alone application and the algorithms can not be adjusted easily. Unlike the gstat package in R that can be extended and then uploaded by anybody. A limitation of R, however, is that it can reach memory use problems if larger rasters or larger quantity of rasters are loaded into R. Visualisation of large maps in R is also not recommended.

So in summary, if you wish to fit your data in a statistically most optimal way and with no limitations on the number of predictors and statistical operations, then you should definitely do it directly in R. If you do not feel confident about using software environment without an interface, then you should try running global Universal kriging in SAGA. However, SAGA does not provide professional variogram modelling options, so a combination of R and SAGA is probably the best idea. In fact, the computation is a bit faster in SAGA than in R and there are no memory limit problems. However,

Table 3.1: Comparison of computing capabilities of some popular statistical and GIS packages (versions in year 2007): ★ — full capability, ⋆ — possible but with many limitations, − — not possible in this package. Commercial price category: I — > 1000 EUR; II — 500-1000 EUR; III — < 500 EUR; IV — open source or freeware. Main application: A — statistical analysis and data mining; B — interpolation of point data; C — processing of auxiliary maps; E — preparation and visualization of final maps. After Hengl et al. (2007b).

Aspect	S-PLUS	R+gstat	SURFER	ISATIS	GEOEas	GSLIB	GRASS	PC Raster	ILWIS	IDRISI	ArcGIS	SAGA
Commercial price category	II	IV	III	I	IV	IV	IV	III	IV	II	I	IV
Main application	A, B	B	B, E	B	B	B	B, C	C	B, C	B, C	B, E	B, C
User-friendly environment to non-expert	★	–	★	★	–	–	⋆	–	★	★	★	★
Quality of support and description of algorithms	⋆	★	⋆	★	★	★	⋆	⋆	★	★	⋆	⋆
Standard GIS capabilities	–	⋆	⋆	–	–	–	★	⋆	★	★	★	★
Standard descriptive statistical analysis	★	★	–	★	⋆	⋆	⋆	–	★	★	★	⋆
Image processing tools (orthorectification, filtering, land surface analysis)	–	–	–	–	–	–	⋆	–	★	★	⋆	★
Comprehensive regression analysis (regression trees, GLM)	★	★	–	⋆	–	–	–	–	–	⋆	–	–
Interactive (automated) variogram modelling	–	★	–	★	–	⋆	–	–	–	★	★	–
Regression-kriging with auxiliary maps	–	★	–	⋆	–	–	★	⋆	⋆	★	–	★
Dynamic modelling (simulations, spatial iterations, propagation, animations)	–	⋆	–	–	–	–	⋆	★	⋆	⋆	⋆	⋆

in SAGA you will not be able to objectively estimate the variogram of residuals or GLS model for deterministic part of variation.

3.7.2 Getting addicted to R

From the above-listed packages, one package needs to be especially emphasized and that is R. Many R users believe that there is not much in statistics that R can not do[12]. Certainly, the number of packages is increasing everyday, and so is the community. There are at least five good (objective) reasons why you should get deeper into R (Rossiter, 2007a):

It is of high quality — It is a non-commercial product of international collaboration between top statisticians.

It helps you think critically — It stimulates critical thinking about problem-solving rather than a *push the button* mentality.

It is an open source software — Source code is published, so you can see the exact algorithms being used; expert statisticians can make sure the code is correct.

It allows automation — Repetitive procedures can easily be automated by user-written scripts or functions.

It can handle and generate maps — R now also provides rich facilities for interpolation and statistical analysis of spatial data, including export to GIS packages and Google Earth.

The main problem with R is that each step must be run via a command line, which means that the analyst must really be an R expert. Although one can criticize R for a lack of an user-friendly interface, in fact, the most power users in statistics never use GUIs (there is the same issue with the SAS procedures). GUI's are fine for baby-steps and getting started, not for a real production work. The whole point is that one can develop a script or program that, once it is right, it can be re-run and it will produce exactly the same results (excluding simulations of course).

Another import aspect we need to consider is R's fitness to work with large dataset. Currently, many pharmaceutical organizations and financial institutions that use R as their main engine, crunching huge amounts of data, so it is an operational tool that can be used for large projects. However, when it comes to GIS data, R still has serious limitations, both to load, display and process large raster maps ($\gg$1M pixels[13]). This problem, as mentioned previously, can be solved by combining R with other (preferably open-source) GIS packages. Colleagues from the Centre for e-Science in Lancaster have been recently developing an R package called MultiR that should be able to significantly speed up R calculations by employing the grid computing facilities (Grose et al., 2006).

Here are some useful tips on how to get addicted to R. First, you should note that you can edit the R scripts in an user-friendly script editors such as Tinn-R or use the package R commander (Rcmdr), which has an user-friendly graphical interface. Second, you should take small steps before you can get into really sophisticated script development.

---

[12]This is probably somewhat biased statement. For example, R is definitively not (yet) fit for image processing (filter analysis, map iterations etc.) and interactive visual exploration of spatial data.

[13]One solution is to manually set the memory limits e.g. `--min-vsize=10M --max-vsize=3G --min-nsize=500k --max-nsize=100M` where Windows starts the `Rgui.exe` (right-click the R shortcut on your desktop and add this line at the end of the `Rgui.exe` address).

Start with some simple examples and then try to do the same exercises with your own data. The best way to learn R is to look at the existing scripts. For example, a French colleague, Romain François, has been maintaining a gallery of R scripts that is dedicated to the nobel goal of getting you addicted to R. Third, if your R script does not work, do not break your head, try to get the book of Chambers and Hastie (1992), Venables and Ripley (2002) and Murrell (2006) or search internet for people with similar problems. Web-resources on R are quite extensive and often you will find all that you need. If nothing from the above helps, try to contact some of the R gurus. However, keep in mind that these are extremely busy people and that they prefer to communicate with you about your problems via some of the R mailing lists[14].

3.7.3   Further software developments

There are still many geostatistical operations that we are aware off, but have not been implemented and are not available to broader public (§2.8.3). What the programmers definitively might consider for future is the refinement of (local) RK in a moving window. This will allow not only better data fitting, but will also allow users to visualize variation in regression (maps of R-square and regression coefficients) and variogram models (maps of variogram parameters). Note that the RK with moving window would need to be fully automated, which might not be an easy task considering the computational complexity. Also, unlike the OK with moving window (Walter et al., 2001), RK has much higher requirements considering the minimum number of observations (at least 10 per predictor, at least 50 to model variogram). In general, our impression is that much of the procedures (regression and variogram modelling) in RK can be automatized and amount of data modelling definitions expanded (local or global modelling, transformations, selection of predictors, type of GLMs etc.), as long as the point data set is large and of high quality. Ideally, user should be able to easily test various combinations of input parameters and then (in real-time) select the one that produces most satisfactory predictions.

The open-source packages open the door to analyses of unlimited sophistication. However, they were not designed with graphical user interface, wizards, nor interactivity typical for commercial GIS packages. Because of this, they are not easily used by nonexperts. There is thus opportunity both for commercial GIS to incorporate RK ideas, or for open-source software to become more user-friendly. Because many statistical techniques can be automated, integration of GIS and statistical algorithms, in the near future, should open the possibility to easily and quickly interpolate dozens of variables by using dozens of predictors.

3.7.4   Towards a system for automated mapping

Geostatistics provides a set of mathematical tools that have been used now over 50 years to generate maps from point observations and to model the associated uncertainty. It has proven to be an effective tool for large quantity of applications ranging from mining and soil and vegetation mapping to environmental monitoring and climatic modelling. Several years ago, geostatistical analysis was considered to be impossible without intervention of a spatial analyst, who would manually fit variograms, decide on the support

---

[14]Think about it — R developers are not employees of R.com! They are not responsible to provide ANY support. If you really want to get some useful feedback then try to ask SMART questions that contribute to the progress of the whole community, and not only to your personal goals. Before posting any questions, make sure you read the R posting guide.

size and elaborate on selection of the interpolation technique. Today, the heart of a mapping project can be the computer program that implements (geo)statistical algorithm that has shown itself to be successful in predicting target values. This leads to a principle of **automated mapping** where the analyst focuses his work only on preparing the inputs and supervising the data processing[15]. This way, the time and resources required to go from field data to the final GIS product (geoinformation) are used in an economical manner.

Automated mapping is still utopia for many mapping agencies. At the moment, each group in the world involved with environmental monitoring runs analysis separately, often without using the right technique, frequently without making the right conclusion, and almost always without considering data/results of adjacent mapping groups. On one side, the amount of field and remote sensing data in the world is rapidly increasing (Table 3.2); on the other side, we are not able to provide reliable information to decision makers in near real-time. It is increasingly necessary that we automate the production of maps (and models) that depict environmental information. In addition, there is an increasing need to bring international groups together and start *"piecing together a global jigsaw puzzle"*[16]. All this proves that automated mapping is an emerging research field and will receive a significant attention in geography and Earth sciences in general (Hiemstra et al., 2007).
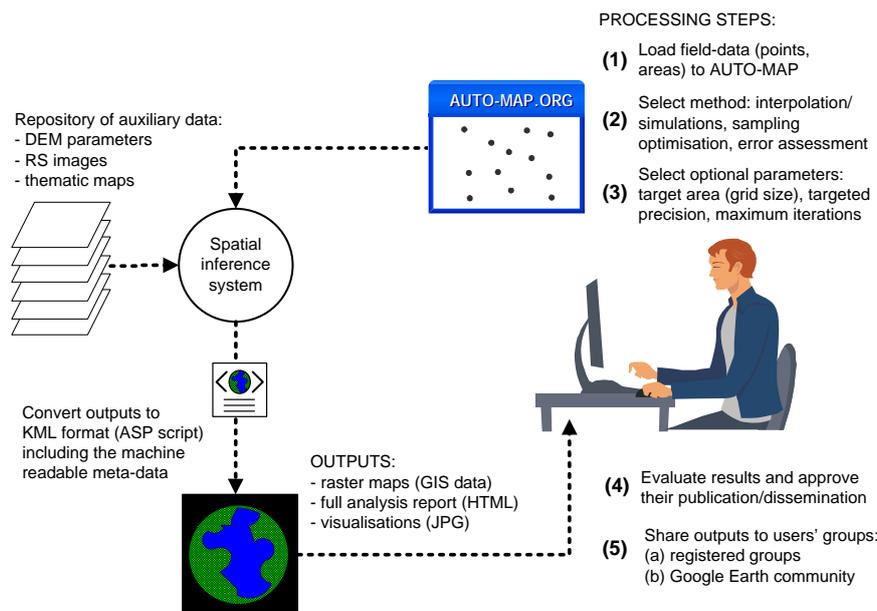
Fig. 3.19: The flow of procedures in auto-map.org: a web-based system for automated predictive mapping using geostatistics. The initial fitting of the models is completely automated, the user then evaluates the results and makes eventual revisions.

The author of this guide with a group of collaborators have already started some preliminary work to design, develop and test a web-based automated mapping system called **auto-map.org**. This web-portal should allow the users to upload their point data and then: (a) produce the best linear predictions, (b) interpret the result of analysis through an intelligent report generation system, (c) allow interactive exploration of the uncertainty, and (d) suggest collection of additional samples — all at click of button.

---

[15]See for example outputs of the INTAMAP project.

[16]Ian Jackson of the British Geological Survey; see also the OneGeology project.

Table 3.2: Some popular sources of (global) auxiliary data sets available at no cost or under an academic licence (e.g. for use in research, environmental protection or for educational uses).

Data set	Description
SRTM DEM	A global Digital Elevation Model produced by the Shuttle Radar Topography Mission (SRTM). The area covered is between 60° North and 58° South. It was recorded by X-Band Radar (NASA and MIL, covering 100% of the total global area) and C-Band Radar (DLR and ASI, covering 40%). The non-public DLR-ASI data is available with a resolution of approximately 30 m (1 arcsec).
MODIS images	MODIS (or Moderate Resolution Imaging Spectroradiometer) is a key instrument aboard the Terra (EOS AM) and Aqua (EOS PM) satellites. The images include the surface reflectance measures, land surface temperatures and emissivity, vegetation indices, thermal anomalies, Leaf Area Index and similar, all at resolution of 250 m. Both satellites are viewing the entire Earth's surface every 1 to 2 days, acquiring data in 36 spectral bands.
SPOT vegetation images	SPOT offers a relatively coarse vegetation-based 10–day images of the whole Earth collected in the period from 1998 until today. Only two bands are available at the moment: NDVI and radiometry images. More detailed multi-spectral images and DEMs can be order from SPOT at a commercial price.
GeoCover LANDSAT scenes	High resolution (15 m) Landsat images for nearly all of the world (years 1990 and 2000) can be downloaded from the NASA's website. The color composites consists from three bands: band 7 (mid-infrared), band 4 (near-infrared) and band 2 (visible green). The images are in the MrSID format and projected in the UTM coordinate system.
ENVISAT images	The ENVISAT satellite is a platform for several instruments adjusted for monitoring of the environmental resources: ASAR, MERIS, AATSR, MWR and similar. The MEdium Resolution Image Spectrometer (MERIS) is used to obtain images of the Earth's surface at temporal resolution of 3–days. The images comprise of 15 bands, all at resolution of 300 m, and can be best compared to the MODIS Terra images. A registration is needed before the data can be ordered/downloaded.
SEVIRI meteorological images	The Meteosat Second Generation (MSG) satellites (from Meteosat-8 onwards) produce SEVIRI 15–minutes images at resolution of 1 km. For environmental applications, the most attractive data set is the High Rate SEVIRI, which consists of 12 spectral channels including: visible and near infrared light, water vapour band, carbon dioxide and ozone bands.
Global Land Cover Facility maps	GLCF is a center for land cover science located at the University of Maryland. Their most known/used product are the Land cover map of the World (14 classes) derived from the AVHRR satellite imagery acquired between 1981 and 1994, Landsat mosaics for large areas and various vegetation maps.
The DMSP lights at night images	Available via the Defense Meteorological Satellite Program (DMSP), which measures night-time light emanating from the earth's surface at 1 km resolution. The lights at night map contains the lights from cities, towns, and other sites with persistent lighting, including gas flares. The filtered annual composites are available from 1992 until today.
Global Administrative Areas	GADM is a database of the location of the world's administrative areas (boundaries). Administrative areas in this database are countries and lower level subdivisions such as provinces and counties. GADM has data for more than 100,000 areas.
Global Climate layers	WorldClim contains grids of interpolated climatic variables for period 1950-2000 using the measurements from >15,000 weather stations: mean, minimum and maximum temperatures, monthly precipitation and bioclimatic variables. All at ground resolution of 1 km. The maps were produced at the University of California, Berkeley.
The One Geology project maps	This is an on-going project with a purpose to produce dynamic digital 1:1 million geological map of the World that will be distributed via a web portal. The first results have been scheduled for the International Geological Congress in Oslo in 2008.

The analysis will be possible via a web-interface and a plugin (e.g. Google Earth), so that various users can access/employ outputs from various mapping projects. All outputs will be coded using the HTML and Google Earth (KML) language. The registered users will be able to update the existing inputs and re-run analysis or assess the quality of maps (Fig. 3.19). A protocol to convert and synchronize environmental variables coming from various countries/themes will be developed in parallel (based on GML/GeoSciML).

There would be many benefits of having a robust, near-realtime automated mapping tool with a friendly web-interface. These are the some important ones:

- the time spent on data-processing would be seriously reduced; the spatial predictions would be available in near real time;

- through a browsable GIS, such as Google Earth, various thematic groups learn how to exchange their data and jointly organize sampling and interpolation[17];

- the cost-effectiveness of the mapping would increase:

  – budget of the new survey projects can be reduced by optimising the sampling designs;

  – a lower amount of samples is needed to achieve equally good predictions;

It is logical to assume that the software for automated mapping will need to be *intelligent*. It will not only be able to detect anomalies, but also to communicate this information to the users, autonomously make choices on whether to mask out parts of the data sets, use different weights to fit the models or run comparison for various alternatives. This also means that a development of such system will not be possible without a collaboration between geostatisticians, computer scientists and environmental engineers.

Many geostatisticians believe that map production should never be based on a *black-box* system. Author of this guide agrees with these views. Although data processing automation would be beneficial to all, analysts should at any moment have the final control to adjust the automated mapping system if needed. To do this, they should have full insight into algorithms used and be able to explore input datasets at any moment.

Important sources:

★ Conrad, O. 2007. SAGA — program structure and current state of implementation. In: Böhner, J., Raymond, K., Strobl, J., (eds.) "SAGA - Analysis and modelling applications", Göttinger Geographische abhandlungen, Göttingen, pp. 39-52.

★ Rossiter, D.G., 2007. Introduction to the R Project for Statistical Computing for use at ITC. International Institute for Geo-information Science & Earth Observation (ITC), Enschede, Netherlands, 136 pp.

★ Murrell, P., 2006. R Graphics. Chapman & Hall/CRC, Boca Raton, FL, 328 pp.

★ Venables, W. N. and Ripley, B. D., 2002. Modern applied statistics with S. Statistics and computing. Springer, New York, 481 pp.

---

[17]The "*profit from your neighbour*" concept.

★ http://www.52north.org — 52° North initiative where ILWIS GIS can be obtained.

★ http://www.saga-gis.org — homepage of the SAGA GIS project.

★ http://www.gstat.org — homepage of the gstat program.

★ Gstat-info mailing list.

★ AI-Geostats' list of software used for statistical spatial data analysis.

4

A geostatistical mapping exercise

4.1 Case study: Ebergötzen

Ebergötzen is 10×10 km study area in the vicinity of the city of Göttingen in Central Germany (51°30'03.16"–51°35'31.45"N; 10°00'28.67"–10°09'15.21"E). This area has been extensively surveyed over the years, mainly for the purposes of developing operational digital soil mapping techniques (Gehrt and Böhner, 2001). The dataset has also been frequently used by the SAGA development team and the SciLands GmbH in many of their demonstrations and documents.

The dataset consists of four groups of GIS layers (Fig. 4.1):

(1.) **Point observations** — the point dataset consists of lab measurements four variables are available: SAND, SILT and CLAY (all expressed as % of mass measured for the 0-30 cm layer of soil) and SOILTYPE (type of soil based on the German classification system). Point observations are allocated in three tables:

- POINTS (300 observations) — the original dataset used to generate predictions;
- CONTROL (300 observations) — a validation dataset used to assess the accuracy of predictions;
- POINTSAL (2937 observations) — the complete dataset with all original observations;

(2.) **Digital Elevation Models**:

- DEM25 — 25 m DEM derived from the topo-maps;
- DEM100 — 100 m DEM from the SRTM mission;

(3.) **LANDSAT image bands** obtained from the Image 2000 & Corine Land Cover 2000 Project. The image consists of seven bands and one panchromatic band:

- LANDIMG — 25 m bands: Band 1 (B, 0.45-0.52), Band 2 (G, 0.53-0.61), Band 3 (R, 0.63-0.69), Band 4 (NIR, 0.78-0.90), Band 5 (MIR, 1.55-1.75), Band 6 (T, 10.40-12.50), Band 7 (MIR2, 2.09-2.35);
- PANIMG — 12.5 m panchromatic image;

(4.) **Geological map** — from 1:50.000 geological map of Germany:

- ■ ZONES: Z1 — Clay and loess, Z2 — Clayey materials, Z3 — Sandy material, Z4 — Silt and sand;

All input raster maps are in ArcInfo `*.asc` format, and the point data (tables) are in a `*.dbf` format. All coordinates are in the official German coordinate system, zone 3 (`germany3`): Transverse Mercator Projection, central meridian is 9°, false easting 3500000, Bessel 1841 ellipsoid with Potsdam datum. The bounding coordinates of the study area are: `XMIN=3570000`, `YMIN=5708000`, `XMAX=3580000`, `YMAX=5718000`. The input raster maps are available in two grid resolutions: 25 m (fine) and 100 m (coarse). These datasets can be accessed from the course homepage. To navigate to the area and get some impression about the type of terrain, you can use the `zones.kml` (borders of geological units) and `points.kml` (location of points dataset) Google Earth layers.

Purpose of this exercise is to interpolate four variables (`SAND`, `SILT` and `CLAY` and `SOILTYPE`), assess the accuracy of produced maps using an independent validation dataset and prepare final layouts. We will start by running some exploratory data analysis, including descriptive statistics and

Fig. 4.1: The Ebergötzen dataset. Courtesy of Gehrt Ernst, the State Authority for Mining, Energy and Geology, Hannover, Germany.

analysis of the point geometry; then we will prepare the auxiliary predictors — various DEM-parameters, satellite-based soil indices and geological strata — and convert them to soil predictive components; we will finally run predictions and simulations in R and export the results to ILWIS, where the final layouts for Google Earth will be prepared. In §4.7, some further testing and analysis is demonstrated using different detail of predictors (25 m vs 100 m DEMs), different density of points (`POINTS` vs `POINTSAL`), and different types of predictors (remote sensing bands vs DEM-parameters).

We will work parallel in all four packages. In many cases, the same operations are available in both SAGA, ILWIS and R, which will be indicated in the text. We will always opt for a package that is most suited for specific analysis. For example, SAGA will be used for processing of DEMs and extraction of land-surface parameters; ILWIS will be used for processing of remote sensing data, principal component analysis and preparation of final layouts; and R will be used for geostatistical analysis.

In most cases, the instructions will be command-based rather than point-and-click operations. The advantage of using scripts in R, ILWIS or SAGA is that every computational step is recorded, and this history can be saved for later use or documentation. Because many steps are automatic, we can also achieve a fully-automated geostatistical mapping where no intervention of analyst is needed.

The computational steps basically follow the generic framework for digital soil mapping described in detail in Hengl et al. (2004a). This typically consists of five steps (see also Fig. 2.10):

(1.) Preparation of the auxiliary maps and target variables.

(2.) Regression modelling.
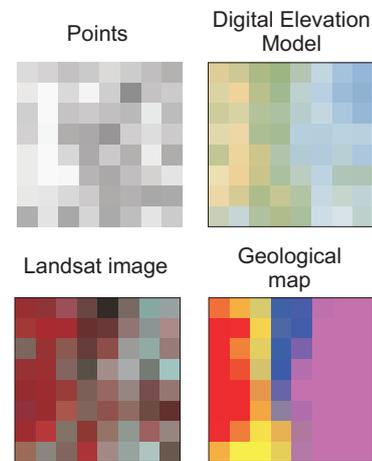
(3.) Variogram modelling of residuals.

(4.) Spatial prediction/simulations.

(5.) Production of final map layouts.

4.2   Data import and preparation of maps

We start by importing the point and raster data to various packages and processing them. First download the ebergotzen.zip file that contains all rasters and tables used in the exercise. Extract the files to some working directory e.g. `d:\geostat\`. You will see all datasets listed in §4.1. In SAGA, ArcInfo ASCII files can be loaded directly using *Modules ↦ File ↦ Grid ↦ Import ↦ Import Arc/Info ASCII*. This will only load the maps in the working memory. All the changes and derivatives you will need to save to the native SAGA grid format `*.sgrd`. Note that you can also do all processing and then export your maps using *Modules ↦ File ↦ Grid ↦ Export ↦ Export Arc/Info ASCII*. Tables (`*.dbf`) can be read directly in SAGA by selecting *File ↦ Table ↦ Load table*. To generate a point map from a table, use *Modules ↦ Shapes ↦ Points ↦ Convert a Table to Points*.

In ILWIS, you will need to import both the rasters and tables to the ILWIS format. To avoid mixing the datasets, I advise you to prepare an additional folder e.g. `d:\geostat\ilwismaps\` where you can keep all ILWIS maps. Note that, once you import a raster map, ILWIS will create three files: (1) a header or object definition file (`*.mpr`), (2) a binary data file that contains all data coded based on the domain type (`*.mp#`), and (3) a georeference file that contains the definition of the grid system. The grid definition file will not have the correct projection system, so you can use the coordinate system available in the zip file. You might also replace the georefence definitions by right-clicking a raster layer and selecting *Properties ↦ Georeference ↦* change the georeference to `grid25m` (or `grid100m` for the 100 m DEM).

4.2.1   The target variables

To import the point data (`*.dbf` table) to R use:

```
library(foreign)
points = read.dbf("points.dbf")
```

where `foreign` is the library for by Minitab, S, SAS, SPSS, Stata, Systat, dBase and similar software packages. This will create an R data object called `points`. The first thing you will probably wish to do is to open this dataset or at least look at it. There are several basic R commands that allow you to get more insight into any datasets, in the case you need to debug a code problem or extend computations:

- `structure` and/or `str` — present the structure of an object, including class, number of elements, variables imbedded etc.

- `class` — informs us about the type of object we are working with;

- `names` — useful to obtain the names of variables and objects within an object;

- `summary` — provides a summary (statistics) view of the data;

Let us first take a look at the structure of this dataset:

```
> str(points)

'data.frame':   300 obs. of  7 variables:
$ ID      : Factor w/ 300 levels "id0003","id0009",..: 1 2 3 4 5 6 7 8 9 10 ...
$ X       : num  3580837 3573361 3580379 3580438 3569344 ...
$ Y       : num  5717700 5708477 5717719 5717685 5715447 ...
$ SAND    : num  18.5 20 18.5 18.5 20 20 15.9 20 15.9 20 ...
$ SILT    : num  67.3 40 67.3 67.3 40 40 63.9 40 63.9 40 ...
$ CLAY    : num  14.1 40 14.1 14.1 40 40 20.2 40 20.2 40 ...
$ SOILTYPE: Factor w/ 13 levels "A","B","D","G",..: 7 2 7 7 9 9 9 9 6 9 ...
- attr(*, "data_types")= chr  "C" "N" "N" "N" ...
```

this means that this a `data.frame` type of an object; it contains 300 observations and 7 variables: `ID`, `X`, `Y`, `SAND`, `SILT`, `CLAY` and `SOILTYPE`. The first and the last variable are of type `Factor`[1] (13 classes) and all other are numeric variables. Note that R has automatically detected which are numerical, which factor-type variables and how many soil types have been observed in total.
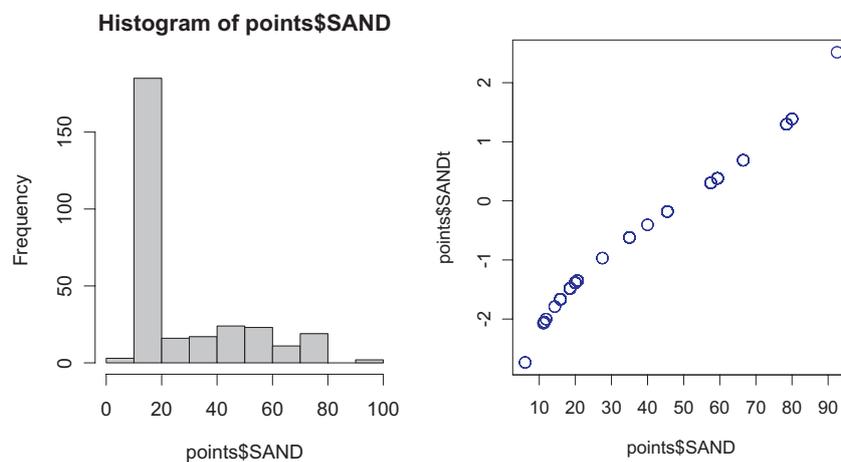
Fig. 4.2: Histogram for `SAND` (left) and the logit transformation plot (right).

Now we can also look at some summary statistical measures for our target variables. First, let us look at the histogram:

```
hist(points$SAND, col="grey")
```

this will produce a plot as shown in Fig. 4.2. Note that the histogram shows that values between 10-20% are the most frequent. Another useful display is the boxplot graph, which will automatically determine the min/max range, 75% quantile range, mean value and depict the outliers. The boxplot for `SAND` definitively shows that the distribution is skewed toward lower values. To see if there are differences between various soil types considering `SAND`, we can invoke a factor-based boxplot by (Fig. 4.3):

```
boxplot(points$SAND~points$SOILTYPE, col="yellow", main="SAND %:
boxplot()s for different soil types")
```

---

[1]A factor is a numeric variable that acts as a categorical variable, i.e. you can do comparisons but you can't do computations with it.

This boxplot should show that the interquantile ranges[2] do not overlap seriously, which means that all soil types differ in their properties.

SAND %: boxplot()s for different soil types
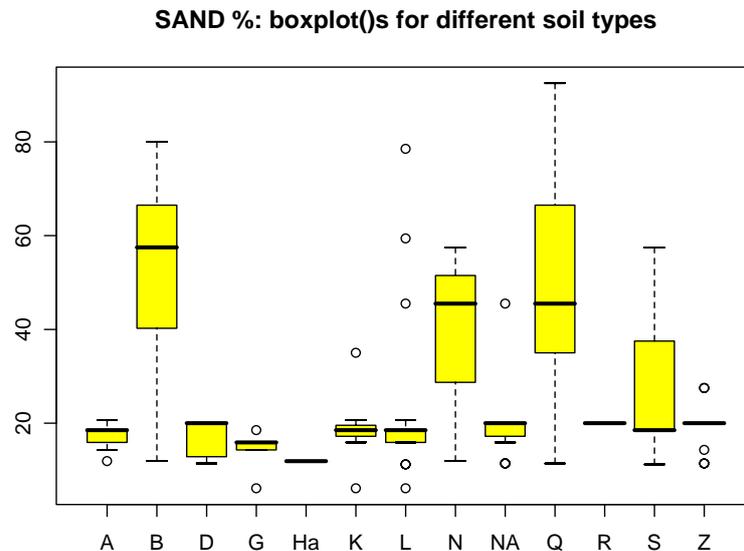
Fig. 4.3: Using the standard boxplot display to assess a thematic overlap between different soil types.

To solve the problem of skewness of distributions of target variables `SAND`, `SILT` and `CLAY`, we can use the **logit transformation** (Hengl et al., 2004a):

$$z^{++} = \ln\left(\frac{z^+}{1-z^+}\right); \qquad 0 < z^+ < 1 \qquad (4.2.1)$$

where z^+ is the target variable standardised to the 0 to 1 range:

$$z^+ = \frac{z - z_{\min}}{z_{\max} - z_{\min}}; \qquad z_{\min} < z < z_{\max} \qquad (4.2.2)$$

and $z_{\min}$ and $z_{\max}$ are the physical minimum and maximum of z (in this case $z_{\min}$=0 and $z_{\max}$=100%). This means that all new predicted values will also be in-between these two limits, which will save us from the problem of generating negative values. The logit transformation in R can be achieved using:

```
points$SANDt = log((points$SAND/100)/(1-(points$SAND/100)))
```

After the transformation, the values of `SANDt` range from -3 to 3, and the predictions can be within the $[-\infty, +\infty]$ range (Fig. 4.2). The histogram of the transformed variable is now closer to normality, and the boxplot shows better symmetry. The normality of the target variable is important because it is a standard requirement for both regression analysis and kriging. If the values of a target variable are skewed around the regression line, this means that the model can lead to over- or under-estimation. Also note that, if you plan to use the logit transformation with your own data, zero measurements need to be replaced with an arbitrary small number — for example the precision[3] of measuring a variable in the laboratory or in the field. Otherwise, if a value of target variable equals

---

[2]The range between the one quarter largest values and the one quarter smallest values.

[3]In analytical chemistry, this value is known as the **detection limit**.

z_{min} or z_{max}, logit transformation will generate `NA`'s and the final predictions will be corrupted.

Now we can convert the table to point maps so we can run some geostatistical analysis with them. This can be achieved using:

```
library(maptools)
coordinates(points)=~X+Y
```

The `coordinates` function attaches the `X` and `Y` columns as coordinates. Although it seems that nothing drastic has changed with the dataset, our data frame has converted to a spatial points data frame, which has much more complex structure:

```
> summary(points)

Object of class SpatialPointsDataFrame
Coordinates:
      min     max
X 3569344 3580980
Y 5707618 5718820
Is projected: NA
proj4string : [NA]
Number of points: 300
Data attributes:
     ID            SAND             SILT             CLAY         SOILTYPE
 id0003 :  1   Min.   : 6.10   Min.   : 5.00   Min.   : 2.50   L     :78
 id0009 :  1   1st Qu.:18.50   1st Qu.:34.30   1st Qu.:14.10   B     :60
 id0052 :  1   Median :20.00   Median :45.00   Median :20.20   S     :35
 id0055 :  1   Mean   :30.24   Mean   :48.16   Mean   :21.57   Q     :33
 id0094 :  1   3rd Qu.:45.50   3rd Qu.:67.30   3rd Qu.:22.10   NA    :24
 id0112 :  1   Max.   :92.50   Max.   :74.60   Max.   :50.00   Z     :17
 (Other):294                                                   (Other):53
```

Note that the coordinate system is currently set as `NA`, which might give us problems if we wish to export it to KML or similar GIS format. To attach the correct coordinate system we use:

```
proj4string(points) = CRS("+init=epsg:31467")
```

which will set the coordinate system using the European Petroleum Survey Group (EPSG) Geodetic Parameter database. The complete coordinate system is defined by an unique ID "31467". For your own area, you can determine the EPSG ID by downloading the MS Access 97 database file (`EPSG_v6_12.mdb`). Open this database and follow the link to *Forms for browsing or data entry/editing*, then select *Coordinate Reference System* and click on the find button. You can browse the coordinate systems in use by search for a country name (e.g. *Germany*) in the Coordinate Reference System form. An example of the complete EPSG Goederic Parameter entry for coordinate system `germany3.crd` can be seen in Fig. 4.4. If you can not find your coordinate system in this database, you can always set the specific parameters manually, e.g. the `germany3.crd` system can also be defined as:

```
proj4string(points)=CRS("+proj=tmerc +lat_0=0 +lon_0=9 +k=1.000
    +x_0=3500000 +y_0=0 +ellps=bessel +towgs84=580.0,80.9,395.3 +units=m")
```
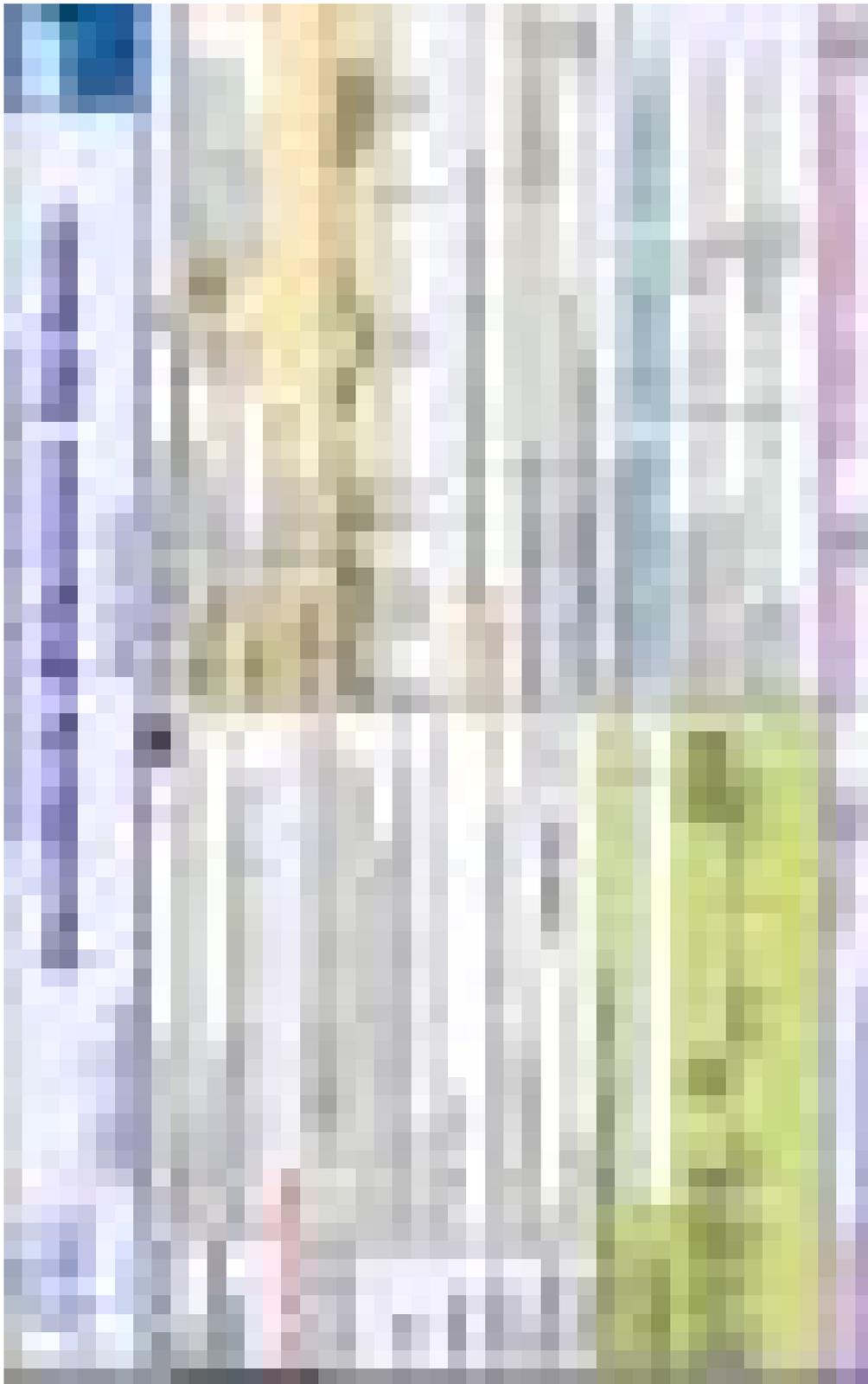
Fig. 4.4: A complete definition of the coordinate system used in the case study Ebergötzen. Obtained from the EPSG Geodetic Parameter database.

where `+proj` is the type of projection, `+lat_0`, `+lon_0` are the latitude and longitude of natural origin, `+x_0`, `+y_0` are false easting and northing, `+ellps` is the name of ellipsoid and `+towgs84` are the datum x, y, z shifts from the WGS84 system.

One you have converted the table to a spatial (point) data frame, you can display it using the `sp` package and the `bubble` plot command:

```
bubble(points["SAND"], scales = list(draw = TRUE), pch=19, col="blue")
```

which will produce a map shown in Fig. 1.7a.

At this stage, there are few more things that we would like to explore prior to regression and variogram modelling. First, we would like to detect the variances of the target variables:

```
var(points[c("SAND", "SILT", "CLAY")])
```

which will produce:

```
          SAND       SILT       CLAY
SAND  422.7256 -310.95110 -111.48449
SILT -310.9511  337.11190  -26.62895
CLAY -111.4845  -26.62895  138.29338
```

This actually produced variances for all combinations of variables. We are interested in the diagonal values: `SAND`=422, `SILT`=337 and `CLAY`=138. Do the same for the transformed variables and you will get the following variances: `SAND`=0.989, `SILT`=0.682 and `CLAY`=0.591.

Second, we can also observe how much are our target variables correlated between each other:

```
cor(points[c("SAND", "SILT", "CLAY")])
```

this will produce:

```
           SAND       SILT       CLAY
SAND  1.0000000 -0.8237127 -0.4610887
SILT -0.8237127  1.0000000 -0.1233293
CLAY -0.4610887 -0.1233293  1.0000000
```

as expected, all three variables are negatively correlated[4]. The highest correlation is between `SAND` and `SILT` (r=-0.82). This does not represent a problem for geostatistical mapping, but is an important measure to validate if also the output maps show a similar correlation.

We can also run a **principal component analysis** on the three variables using:

```
pc.text = princomp(~SANDt+SILTt+CLAYt, cor=T, data=points)
```

which will generate three components and summary statistics. In this case, the first component explains 69.2% of variation, second 30.0% and the last component only 0.7%. The results of principal component analysis can be visualized using:

---

[4]These are **compositional variables** — the values are connected by definition. In the case of the three texture fractions, a value of any variable equals 100 less sum of the other two.
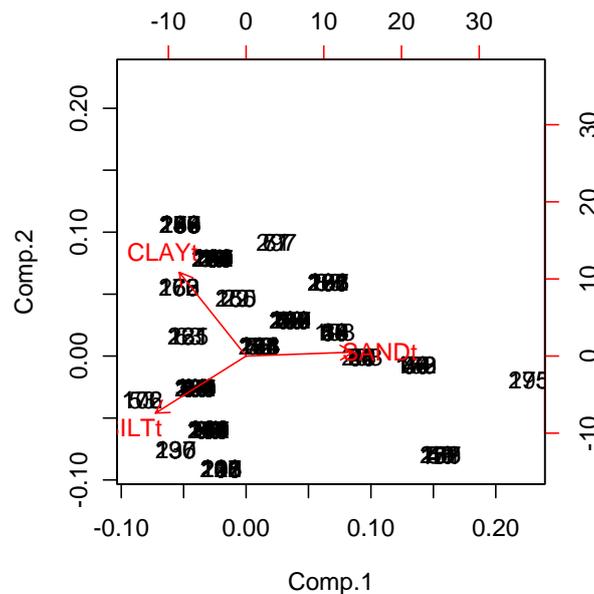
Fig. 4.5: The biplot graph for principal component analysis of `SANDt`, `SILTt`, `CLAYt`.

```
biplot(pc.text)
```

which will produce a standard biplot graph that displays the first two components, original variables represented with arrows and transformed observations. Again, the biplot graph (Fig. 4.5) confirms that all variables are negatively correlated and equally important to explain the overall variability in the data.

From the biplot display, we can also observe that many numbers are in fact overlapping, which makes them hard to read. It seems that there are only 19 clusters of observations in the two dimensional space, although we know that there are 300 observations in this dataset (see also the groupings in the variogram cloud in Fig. 1.7b). Going back to the origin of this dataset we need to note that the sand, silt and clay values have been determined by using the so-called *texture by hand* method[5]. The literature (Skaggs et al., 2001) reports that the this technique can be used to determine the content of soil earth fractions only to an accuracy of ± 5–10%. This means that we should not plan to map any of the texture fractions to a precision better than $\pm 5\%$, because it would exceed the measurement error.

4.2.2   Auxiliary maps — predictors

We will use the auxiliary predictors from three sources of data: (a) remote sensing-based indices, (2) DEM-based parameters and (3) geological units (strata or polygons). Note that, because we wish to use the Landsat bands to map soil texture and distribution of soil types, we need to use some relative indices that have shown to be related with soil properties. We derive three soil-related indices[6] based on the Landsat TM imagery:

---

[5]A surveyor distinguishes to which of the 32 texture classes a soil samples belongs to, and then estimates the content of fractions. E.g. texture class `St2` has 10% clay, 25% silt and 65% sand.

[6]You can derive these by using the map calculation syntax in ILWIS. Follow the ILWIS script available in the zip file for complete details.

(1.) Grain Size Index (`GSI`) (Xiao et al., 2006):

$$\texttt{GSI} = \frac{\texttt{R} - \texttt{B}}{\texttt{R} + \texttt{G} + \texttt{B}} \tag{4.2.3}$$

(2.) Clay Index (`CI`):

$$\texttt{CI} = \frac{\texttt{MIR}}{\texttt{MIR2}} \tag{4.2.4}$$

(3.) and Normalized Difference Vegetation Index (`NDVI`):

$$\texttt{NDVI} = \frac{\texttt{NIR} - \texttt{R}}{\texttt{NIR} + \texttt{R}} \tag{4.2.5}$$

From the DEM, we can derive the following five parameters:

(1.) Slope gradient in % (`SLOPE`);

(2.) SAGA Wetness index (`TWI`);

(3.) Incoming solar radiation (`SOLAR`);

(4.) Plan curvature (`PLANC`);

(5.) Profile curvature (`PROFC`);

`SLOPE` and `TWI` reflect the erosion/deposition potential of a terrain and can be derived in SAGA. The SAGA `TWI` is based on the modified catchment area that is estimated iteratively to better represent large floodplains. Incoming solar radiation [kWh/m^2] you can derive also in SAGA for a period of one year using the default settings. It represents the climatic conditions important for soil formation. Plan and profile curvature you can derive in ILWIS using the scripts available at http://spatial-analyst.net. These two measures of local morphometry can be used to represent hydrological factors of soil formation.

The final group of predictors are the four indicator maps (`Z1`, `Z2`, `Z3`, `Z4`) showing geological zones. These maps were produced by rasterizing a polygon map (`ZONES`) and then converting it to indicators[7], which can be done in ILWIS by using:

```
Z1{dom=value.dom;vr=0:1:1}=iff(ZONES="Z1",1,0)
```

which makes a raster map with a value domain (numeric variable) showing value 1 at the location of `Z1` and showing 0 for all other locations.

4.2.3   Assessment of the point geometry and sampling quality

As noted in the preface of this handbook, no geostatistician can promise high quality products without quality input point samples. To assess how representative and consistent is our input data, we can run some basic analysis of the point geometry and then overlap the points with predictors to see how well are the environmental features represented.

---

[7]As we will see later on, in R, there is no need for this manual conversion because R automatically generates indicators from factor variables.

Start with point pattern analysis in ILWIS by selecting *Operations ↦ Statistics ↦ Points ↦ Pattern analysis*. This will calculate a probability of finding the 1-6 points and probability of finding all points at certain distance (Boots and Getis, 1988). The output graph can be seen in Fig. 4.6. From the output of this analysis, we are interested in three numbers: (1) 0.5 probability distance of finding one neighbour[8], (2) 1.0 probability distance of finding one neighbour, and (3) 1.0 probability distance of finding all neighbours. The last measure can be use used to represent the extent of the area, i.e. to input parameters for the definition of the standard initial variogram (see page 3.4.1).

Fig. 4.6: Results of point pattern analysis for the `points` data set.

In the case of the `points` data set, we can see that the 0.5 probability distance of finding one neighbour is at about 250 m, 1.0 probability distance of finding one neighbour is at 950 m and the 1.0 probability distance of finding all neighbours is at 13.6 km. To ensure that in less than 5% of grid cells points do not fall into the same grid cell, we should select a grid cell size of at least (Hengl, 2006):

$$\Delta\mathbf{s} \leq 0.25 \cdot \sqrt{\frac{A}{n}} = 0.25 \cdot \sqrt{\frac{10^8}{300}} \approx 150 \text{ m} \tag{4.2.6}$$

Following the cartographic rule, used in soil mapping, that there should be at least one (ideally four) observation per 1 cm^2 of the map gives us the approximate effective scale number for this dataset:

$$SN = \sqrt{4 \cdot \frac{A}{n}} \cdot 10^2 \quad \dots \quad SN = \sqrt{\frac{A}{n}} \cdot 10^2 \tag{4.2.7}$$

and in the case of the points data set, there are 222 points[9] spread over an area of 100 km^2, which shows that the effective scale of these maps will be between 1:125k and 1:75k. We can also estimate the suitable grid cell size using (Fig. 4.7):

$$\Delta\mathbf{s} = 0.0791 \cdot \sqrt{\frac{A}{n}} \approx 50 \text{ m} \tag{4.2.8}$$

Note that our original predictors are available in somewhat finer resolution (25 m), so we will try to produce maps using regression-kriging which are at two times better scale[10] than our inspection density.

Next we want to evaluate the quality of the point sample considering how well does it represent the study area. We consider two aspects of sampling quality: (a) point density and (b) representation of feature space. The point density can be run in ILWIS by using:

```
density=MapRasterizePointCount(points.mpp,grid2km.grf,1)
```

---

[8]This is a more objective measure of clustering than the Mean Shortest Distance.

[9]Some of the 300 points fall outside the sampling area, which can be nicely seen by using the SAGA GIS.

[10]In geoinformation science this is referred to as *downscaling* or *disaggregation* (Hengl, 2006).
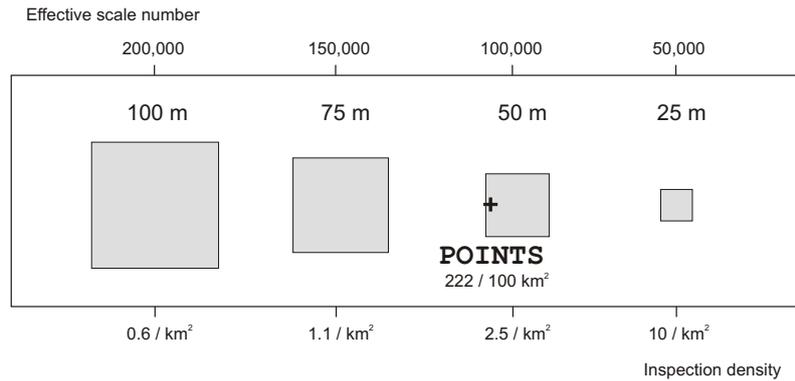
Fig. 4.7: Possible cell sizes for the Ebergötzen case study.

where `MapRasterizePointCount` is the ILWIS function that will count number of points within a grid node (`grid2km`). Assuming that there should be at least $n_{\min}=10$ points within an equal size sub-area (cluster), we have derive the cell size of the clusters to assess the inspection density as:

$$\Delta \mathbf{S} = \sqrt{\frac{A}{n} \cdot n_{\min}} \qquad (4.2.9)$$

which gives approximately 2-kilometer grid. The ILWIS density map for these clusters is shown in Fig. 4.8, left. Note that, to derive the actual values per km^2 we had to divide the density per grid with the area of each grid. You could generate several random designs and then compare if the actual design is significantly different from the completely random design.
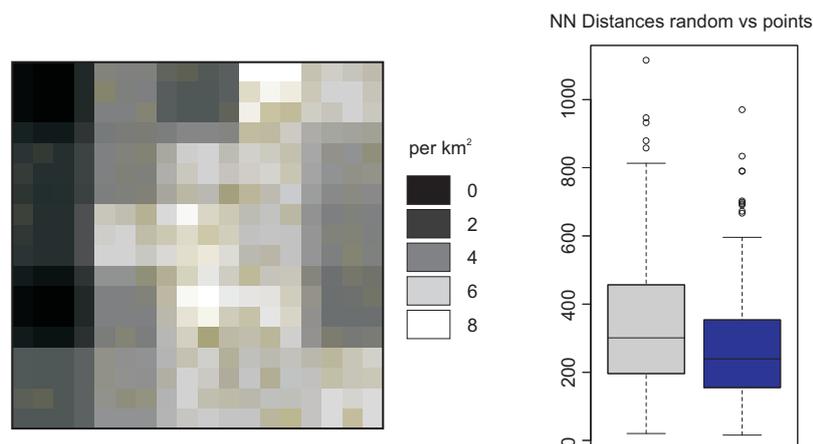
Fig. 4.8: Density of points on a 2×2 km grid (left) and a Boxplot comparison of distances for random point pattern and the `points` data set (right).

A more sophisticated point pattern analysis can be achieved by using the R package spatstat. This has an extensive collection of methods to generate sampling designs, run analysis on distances and analyze spatial autocorrelation structure in the data (Baddeley and Turner, 2005). Just for a comparison, we can generate a random point pattern,

derive distances between the nearest neighbours and then compare the distributions of the distances:

```
library(spatstat)
rand.points=rpoint(300, win=as.owin(c(points@bbox[1],
          points@bbox[3],points@bbox[2],points@bbox[4])))
plot(rand.points)
dist.rpoints=nndist(rand.points$x, rand.points$y)
dist.points=nndist(points@coords[,1], points@coords[,2])
```

where `rpoint` is a `spatstat` method to generate a random point pattern, `win` defines the bounding coordinates, and `nndist` will calculate a matrix of nearest neighbour distances in a point pattern.

We can display the summary statistics for distances derived for a random point pattern and our input points by putting two boxplots next to each other[11] (Fig. 4.8, right):

```
boxplot(nndist.rpoints, at = 1:1 -0.25, col="grey",
        main="Distances random vs points")
boxplot(nndist.points, at = 1:1 +0.25, col="blue", add=TRUE)
```

In the case of the points data set, we can observe that the average spacing to the nearest neighbour is in the range [155, 354], which is somewhat less then for the simulated random design [196, 457]. Although the boxplot comparison shows that the `points` data set is somewhat more clustered, it seems that our data set has a very similar distribution of distances as the random design.

Next, we want to assess spreading of the points in the feature space, for which we need to import the rasters in R. This can be using the rgdal package:

```
library(rgdal)
predictors = readGDAL("ilwis/DEM25.mpr")
```

which will do:

```
ilwis/DEM25.mpr has GDAL driver \textsf{ILWIS}
and has 400 rows and 400 columns
Closing GDAL dataset handle 0x00242b80...  destroyed ... done.
```

which means that R has recognized the ILWIS GIS format and imported a dataset of size 400×400 grids. Other grid layers we can import using:

```
predictors$SLOPE = readGDAL("ilwis/SLOPE.mpr")$band1
predictors$PLANC = readGDAL("ilwis/PLANC.mpr")$band1
predictors$PROFC = readGDAL("ilwis/PROFC.mpr")$band1
predictors$SOLAR = readGDAL("ilwis/SOLAR.mpr")$band1
predictors$TWI = readGDAL("ilwis/TWI.mpr")$band1
predictors$GSI = readGDAL("ilwis/GSI.mpr")$band1
predictors$CI = readGDAL("ilwis/CI.mpr")$band1
predictors$NDVI = readGDAL("ilwis/NDVI.mpr")$band1
predictors$Z1 = readGDAL("ilwis/Z1.mpr")$band1
predictors$Z2 = readGDAL("ilwis/Z2.mpr")$band1
predictors$Z3 = readGDAL("ilwis/Z3.mpr")$band1
predictors$Z4 = readGDAL("ilwis/Z4.mpr")$band1
```

---

[11]In this case, by adding an argument `add=TRUE`, the second boxplot will be added to the current plot, shifted for -0.25 of the bar width to the right.

In this case, we need to specify that we wish to import only the first band (`$band1`) of the ILWIS raster map, otherwise the command would overwrite the whole `predictors` data set. Note that it can take time until R imports huge rasters so have this on your mind. We can now look at the structure of the data set `predictors`:

```
> str(predictors)
Formal class 'SpatialGridDataFrame' [package "sp"] with 6 slots
  ..@ data       :'data.frame': 160000 obs. of  13 variables:
  .. ..$ SLOPE: num [1:160000]  6.1 18.0 32.5 40.8 38.2 ...
  .. ..$ PLANC: num [1:160000] -0.440 -0.499 -0.841 -1.588 -2.391 ...
  .. ..$ PROFC: num [1:160000]  0.467  0.427  0.211 -0.076 -0.393 ...
  .. ..$ SOLAR: num [1:160000] 1299 1285 1249 1148  806 ...
  .. ..$ TWI  : num [1:160000]  9.05  9.12  9.76  9.28 10.76 ...
  .. ..$ GSI  : num [1:160000] -0.137 -0.128 -0.121 -0.129 -0.136 ...
  .. ..$ CI   : num [1:160000] 2.32 2.27 2.27 2.29 2.27 ...
  .. ..$ NDVI : num [1:160000] 0.608 0.602 0.602 0.597 0.586 0.58 0.589 ...
  .. ..$ Z1   : num [1:160000] 1 1 1 1 1 1 1 1 1 1 ...
  .. ..$ Z2   : num [1:160000] 0 0 0 0 0 0 0 0 0 0 ...
  .. ..$ Z3   : num [1:160000] 0 0 0 0 0 0 0 0 0 0 ...
  .. ..$ Z4   : num [1:160000] 0 0 0 0 0 0 0 0 0 0 ...
  .. ..$ DEM25: num [1:160000] 383 380 374 363 351 ...
  ..@ grid       :Formal class 'GridTopology' [package "sp"] with 3 slots
  .. .. ..@ cellcentre.offset: Named num [1:2] 3570013 5708013
  .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
  .. .. ..@ cellsize         : num [1:2] 25 25
  .. .. ..@ cells.dim        : int [1:2] 400 400
  ..@ grid.index : int(0)
  ..@ coords     : num [1:2, 1:2] 3570013 3579988 5708013 5717988
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : NULL
  .. .. ..$ : chr [1:2] "x" "y"
  ..@ bbox       : num [1:2, 1:2] 3570000 5708000 3580000 5718000
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "x" "y"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
  .. .. ..@ projargs: chr " +proj=stere +lat_0=0 +lon_0=9 +k=1.000000
  +x_0=3500000 +y_0=0 +ellps=bessel +datum=potsdam +units=m
```

which shows that predictors are a grid data frame with 13 bands. By typing:

```
>object.size(predictors)
```

you will notice that this data set occupies almost 17MB of the memory for the 25 m resolution maps (160,000 grid nodes per band). For practical reasons, we will pack the indicators and prepare a single spatial layer with factors:

```
predictors$ZONES = as.factor(apply(as.data.frame(predictors)[c("Z1",
"Z2", "Z3", "Z4")],+ 1, function(x) which(x == 1)))
```

which will produce a new factor-type band with classes "1", "2", "3" and "4". We can also copy the name of the first band so that its name in the grid pack is DEM25, and then delete it from the memory:

```
predictors$DEM25 = predictors$band1
predictors$band1 = NULL
```

We also need to attach the correct coordinate system by using:

```
proj4string(predictors) = CRS("+init=epsg:31467")
```

Now, we can overlay the points over grids and to obtain the values of predictors at point locations:

```
predictors.ov = overlay(predictors, points)
```

which will create a new point data frame with the same coordinates as `points` but with attached attributes of the `predictors` grids. We need to copy the values of predictors to our target data set by:

```
points$DEM25 = predictors.ov$DEM25
points$TWI = predictors.ov$TWI
points$GSI = predictors.ov$GSI
points$ZONES = predictors.ov$ZONES
```

and now we can do some preliminary analysis to investigate how well are the environmental features represented with our point samples. For example, we can compare the histograms of the maps versus the histograms of values at point locations, e.g. by using a back to back histogram[12]:

```
library(Hmisc)
options(digits=1)
DEM25.histbb=histbackback(points$DEM25, predictors$DEM25, prob=TRUE)
```

This will produce two histograms next to each other so that we can visually compare how well do the samples represent the original feature space of the raster maps (Fig. 4.9). In the case of the `points` data set, we can see that the samples are misrepresenting (a) higher elevations, (b) tops of hills/ridges and (c) areas of low `GSI` (forests).
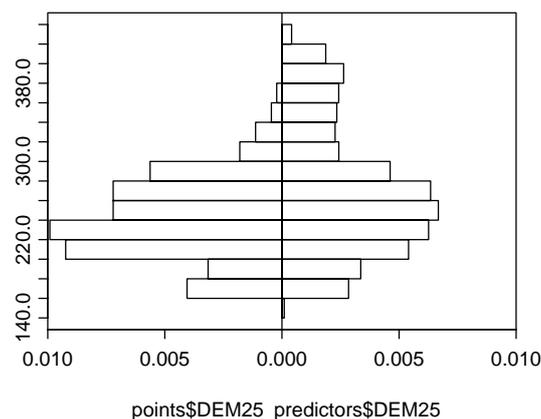
Fig. 4.9: Histogram for sampled values of `DEM` (300) versus the histogram of the raster map (all raster nodes).

This is no surprise if we know that the surveyors have focused on sampling only agricultural soils and have purposely omitted forest areas. We can actually test if the histograms of sampled variables are significantly different from the histograms of original raster maps e.g. by using a non-parametric test such as the Kolmogorov-Smirnov test:

---

[12]This requires installation of the package Hmisc.

```
ks.test(DEM25.histbb$left, DEM25.histbb$right)
```

which shows that the two histograms have significantly different distributions (D=0.4, p-value=0.1813). The same discrepancy can be observed for TWI (p-value=0.4676) and GSI (p-value=0.2154). Another test that you might do to compare the histograms is to run the correlation test[13]:

```
cor.test(DEM25.histbb$left, DEM25.histbb$right)
```

which shows that the two distributions are in fact significantly correlated for DEM (r=0.91) and TWI (r=0.821), while the correlation for the GSI is insignificant (r=0.302).

In the last test we will assess whether the sampling density within different geological units (ZONES) is consistent. First, we look at how many points fall into each zone:

```
>summary(points$ZONES)
   1    2    3    4 NA's
   4   43   34  141   78
```

then we look at the size of the zones in pixels (each 25 m^2):

```
>summary(predictors$ZONES)
     1      2      3      4
36182  28872  22004  72942
```

so now we can derive the observed inspection density using:

```
inspdens.obs=summary(points$ZONES)[1:4]/(summary(predictors$ZONES)[1:4]*0.025^2)
inspdens.exp=rep(2.22,4)
```

which finally gives us the following table:

```
          1     2     3     4
obs: 0.177 2.383 2.472 3.093
exp: 2.22  2.22  2.22  2.22
```

which can also be compared by using the Kolmogorov-Smirnov test:

```
ks.test(inspdens.obs, inspdens.exp)
```

In this case, we see that inspection density is also significantly inconsistent considering the map of ZONES, which is not by chance (D=0.75, p-value=0.2106). We could also run similar analysis for land cover types or any other polygon-based predictors.

So in summary, we can conclude for the points dataset that:

- the average distance to the nearest neighbour is 240 m and the extent of the area is 13.6 km;

- this point dataset is suitable for digital soil mapping at scales between 1:115k and 1:60k;

- the sampling intensity is 3 points per km^2, which corresponds to a grid cell size of about 50 m;

---

[13]Also known as the test of no correlation because it computes t-value for correlation coefficient being equal to zero.

- the sampling density varies in geographical space — at least 8% of the area has been ignored and over 30% of the area is sampled with a much lower inspection density;

- the sampling is unrepresentative considering the maps of `TWI` and `GSI` — especially the forest areas and hilltops have been systematically omitted;

- the inspection density misrepresents zone `Z1` and somewhat over-represented zone `Z4`;

These result do not mean that this data set is not suitable to generate maps, but they indicate that it has several limitations considering the representativeness, independency and consistency requirements. These limitations will reflect on the prediction variance, as we will see later on in §4.5.

4.2.4 Pre-processing of the predictors

Now that we have finished screening the target variables and assessing the quality of the sampling plan, we can focus on preparing the predictors — raster maps that will be used to explain the deterministic part of variation in our variables. There are 13 maps in total that will be used to interpolate our soil variables: `DEM25`, `SLOPE`, `PLANC`, `PROFC`, `TWI`, `SOLAR`, `GSI`, `CI`, `NDVI`, `Z1`, `Z2`, `Z3` and `Z4`.

Before we can use these 13 maps to run regression analysis, we need to account for one problem that might influence our further analysis — the **multicolinearity effect** or overlap of information in the predictors. Typically, the assumption of multiple linear regression is that the predictors are independent variables (Neter et al., 1996). We can easily see that the predictors we plan to use to map soil variables are not independent. In ILWIS, you can combine all maps to a map list and then select *Operations* ↦ *Statistics* ↦ *Maplist* ↦ *Correlation Matrix* and this will give you a $p \times p$ matrix with correlation coefficients. As you can see from Fig. 4.10, several predictors are highly correlated — especially `CI` and `NDVI` (0.96), `GSI` and `DEM` (-0.58), `TWI` and `SLOPE` (-0.64), but also `DEM` and `ZONES` and `GSI` and `ZONES`.

An advisable thing to do to reduce the multicolinearity effect is to run a principal component analysis and then, instead of using the original predictors, use the transformed components that are absolutely independent. Although the principal component analysis is possible in R (see also page 94), for large maps it might be computationally more efficient to run this analysis in ILWIS. Before deriving the predictive components in ILWIS, it is advisable to convert all input rasters to the same binary scale (0-255 values). This can be achieved by using *Operations* ↦ *Image processing* ↦ *Stretch* ↦ *Linear Stretch*. Once you converted all 13 predictors to image domain, you can pack them together in ILWIS by typing:

```
crmaplist predictors DEM25img.mpr SLOPEimg.mpr ... Z4img.mpr
```

Now that you have all predictors packed as a map-list, you can extract their principal components by using:

```
SPC.mat = MatrixPrincComp(predictors, 13)
```

This will create a new map-list called SPC and a matrix object with the same name. The matrix object carries the information about the PC coefficients and percentage of variance explained by each SPC band. You will notice that `SPC1` explains already
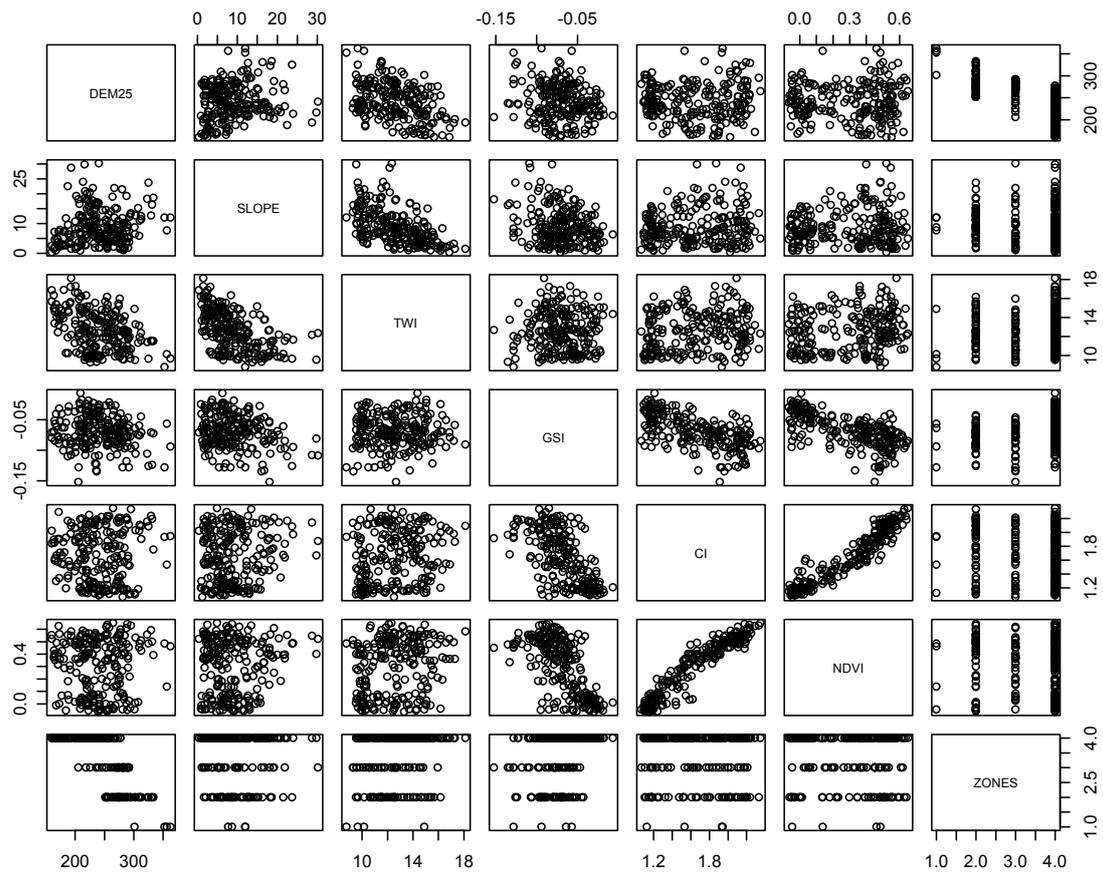
Fig. 4.10: Cross-correlation plots in R derived using the values of predictors estimated at 222 point samples.

46.4%, SPC2 21.1%, SPC3 15.4% and so on. The first five components already explain >95% of the total variance in the data, which proves that PCA is a powerful way to reduce the number of predictors. You can also now visually explore the SPCs in ILWIS to see which general features are represented by which component (Fig. 4.11). For example, it seems that the SPC1, SPC2 and SPC3 jointly represent geomorphometric stratification of the area (ZONES, DEM), SPC4 represents changes in reflectance values (NDVI and CI), SPC5 represents hydrological nature of terrain (TWI) etc. In summary, the original predictors have been converted to independent (mixed) environmental factors. The final components shows less and less variation and often represent the noise in the data. The last component shows no variation at all (due to the rounding effect in ILWIS), so it is better to exclude it from future analysis.

The predictors are now ready to be used as auxiliary maps to improve spatial predictions. Now we can import the SPCs from ILWIS to R by using:

```
SPC = readGDAL("ilwis/SPC_1.mpr")
SPC$SPC2=readGDAL("ilwis/SPC_2.mpr")$band1
...
SPC$SPC12=readGDAL("ilwis/SPC_12.mpr")$band1
```

Note that you can also display the predictors in R using the spplot command, however, this is not recommended for larger rasters ($\gg 10^6$ pixels). You should instead
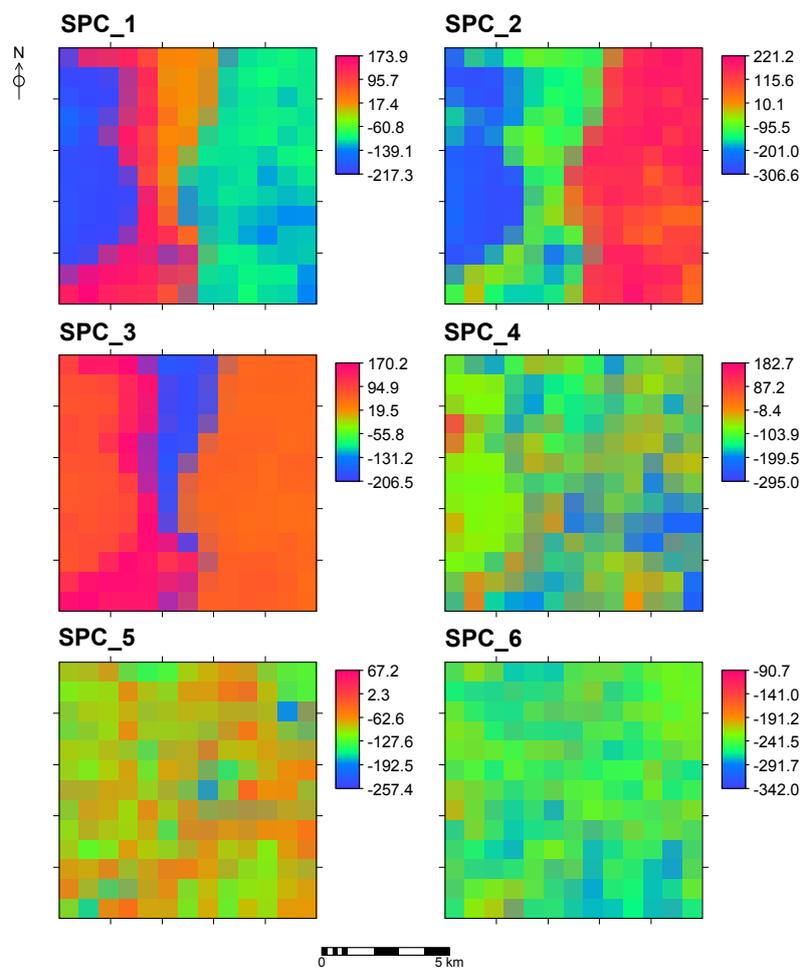
Fig. 4.11: Soil Predictive Components used to predict soil variables (SPCs). Note that the components now show both continuous and discrete changes.

use ILWIS or SAGA to visually explore rasters and R just for the analysis and generation of graphs and variograms.

4.3   Regression modelling

4.3.1   Multiple linear regression

Now that we have imported the 12 SPCs in R, we can try to run some regression analysis and see if the predictors can be used to explain the variation of target variables. Before we can run any analysis, we need to estimate the values of predictors at sampling locations. The original attribute table will then be extended by 12 columns, which will allow us to run regression and variogram analysis and fit the spatial prediction model (Fig. 4.12).

First we overlay the points and rasters by using the sp package:

```
points.ov = overlay(SPC, points)
```
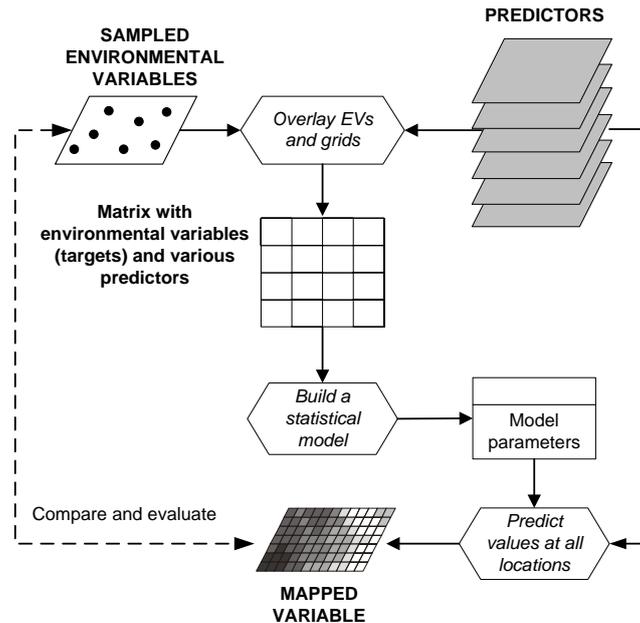
Fig. 4.12: Schematic example of key operations in R used to estimate and apply a (spatial) prediction model by overlaying points and rasters.

this will produce a new point data frame with attributed values of predictors. To simplify the analysis, we need to copy these values to our original `points` data set:

```
points$SPC1 = points.ov$SPC1
points$SPC2 = points.ov$SPC2
...
points$SPC12 = points.ov$SPC12
```

If you now look at the structure of the points data set, you will see that there the initial number of attributes has been extended. You will also notice that some locations are outside the bounding coordinates of the grids, so a `NA` value is attached to them.

Let us now fit a multiple linear regression model:

```
sand.lm = lm(SANDt~SPC1+SPC2+SPC3+SPC4+SPC5+SPC6+SPC7+SPC8+SPC9+SPC10
+SPC11+SPC12, points)
```

where `lm` is the generic R method to fit a linear model and `sand.lm` is the output data frame. If we look at its structure, we will notice that it consists of many elements:

```
> names(sand.lm)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "na.action"     "xlevels"       "call"          "terms"
[13] "model"
```

each of this has a substructure and elements at the lowest level. For example, we can look at the histogram of residuals by typing:

```
hist(sand.lm$residuals, col="grey")
```

or plot the predicted values versus measured values using:

```
sel = !is.na(points$SPC1)
plot(points[sel,]$SANDt, sand.lm$fitted.values)
```

In practice, we are mostly interested is the summary output of the model:

```
> summary(sand.lm)

Call:
lm(formula = SANDt ~ SPC1 + SPC2 + SPC3 + SPC4 + SPC5 + SPC6 +
    SPC7 + SPC8 + SPC9 + SPC10 + SPC11 + SPC12, data = points)

Residuals:
   Min     1Q Median     3Q    Max
-2.040 -0.443 -0.100  0.395  2.942

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.645556   1.142926   -1.44  0.15143
SPC1         0.000853   0.000757    1.13  0.26107
SPC2         0.003158   0.000942    3.35  0.00095 ***
SPC3        -0.007114   0.000592  -12.01  < 2e-16 ***
SPC4        -0.001721   0.000870   -1.98  0.04928 *
SPC5         0.007907   0.001190    6.64  2.6e-10 ***
SPC6        -0.000465   0.002129   -0.22  0.82745
SPC7         0.001544   0.002739    0.56  0.57365
SPC8         0.001371   0.003049    0.45  0.65341
SPC9         0.000849   0.003467    0.25  0.80665
SPC10        0.008015   0.003573    2.24  0.02595 *
SPC11       -0.008728   0.004918   -1.77  0.07743 .
SPC12       -0.002683   0.007137   -0.38  0.70741
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.739 on 209 degrees of freedom
  (78 observations deleted due to missingness)
Multiple R-Squared: 0.534,     Adjusted R-squared: 0.508
F-statistic:   20 on 12 and 209 DF,  p-value: <2e-16
```

which shows that the model explains 51% of variability and is statistically significant. By looking at the specific t-values of coefficients, we can also infer about which predictors are the most significant. In this case, only `SPC5`, `SPC3` and `SPC2` are statistically significant at <0.001 probability level.

4.3.2   Step-wise selection of predictors

A further useful step is to kick-out many insignificant predictors by using the step-wise regression:

```
fsand = step(sand.lm)
```

If we now look at the summary output, we will notice that the initial model has been reduced to 6 predictors:

```
>summary(fsand)
Call:
lm(formula = SANDt ~ SPC2 + SPC3 + SPC4 + SPC5 + SPC10 + SPC11,
    data = points)

Residuals:
   Min    1Q Median    3Q    Max
-2.108 -0.423 -0.103  0.368  2.974

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.511501   0.874569   -1.73    0.085 .
SPC2         0.002233   0.000538    4.15  4.7e-05 ***
SPC3        -0.006771   0.000506  -13.38  < 2e-16 ***
SPC4        -0.001117   0.000735   -1.52    0.130
SPC5         0.007879   0.001125    7.00  3.2e-11 ***
SPC10        0.007976   0.003494    2.28    0.023 *
SPC11       -0.008198   0.004661   -1.76    0.080 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.733 on 215 degrees of freedom
  (78 observations deleted due to missingness)
Multiple R-Squared: 0.529,       Adjusted R-squared: 0.516
F-statistic: 40.3 on 6 and 215 DF,  p-value: <2e-16
```

The R-square is now slightly better and almost all coefficients are highly significant. We can observe the individual correlation plots by:

```
par(mfrow=c(2, 2))
scatter.smooth(points$SPC3, points$SANDt, span=9/10)
scatter.smooth(points$SPC5, points$SANDt, span=9/10)
scatter.smooth(points$SPC2, points$SANDt, span=9/10)
scatter.smooth(points$SPC10, points$SANDt, span=9/10)
```

which will produce a plot as show in Fig. 4.13. At this stage, it might be advisable to cross check if these correlation plots fit our empirical knowledge about the study area. This is just to avoid some accidental correlations or artifacts. In this case, the value of SAND obviously jumps if we get inside the Z3 (geological unit "*sandy material*"), and if we are at relatively dry terrain positions (low TWI). It seems that the overall best predictors of the distribution of texture fractions are Z3 and TWI. This, in general, fits our expectations. Note also that the correlation plot (Fig. 4.13) between SPC2 and SAND indicates a non-linear relationship. At this stage, the predictors have already been transformed, so we will proceed with using a linear model and hope that the residuals can be still fitted using kriging.

We can repeat the same operations also for SILTt and CLAYt and conclude:

- for SANDt, the model explains 51% of variability, the best predictors are SPC3 and SPC5, the residuals are normally distributed;

- for SILTt, the model explains 50% of variability, the best predictors are SPC3 and SPC5, the residuals are normally distributed;

- for CLAYt, the model explains 48% of variability, the best predictors are SPC3 and SPC1, the residuals are normally distributed;
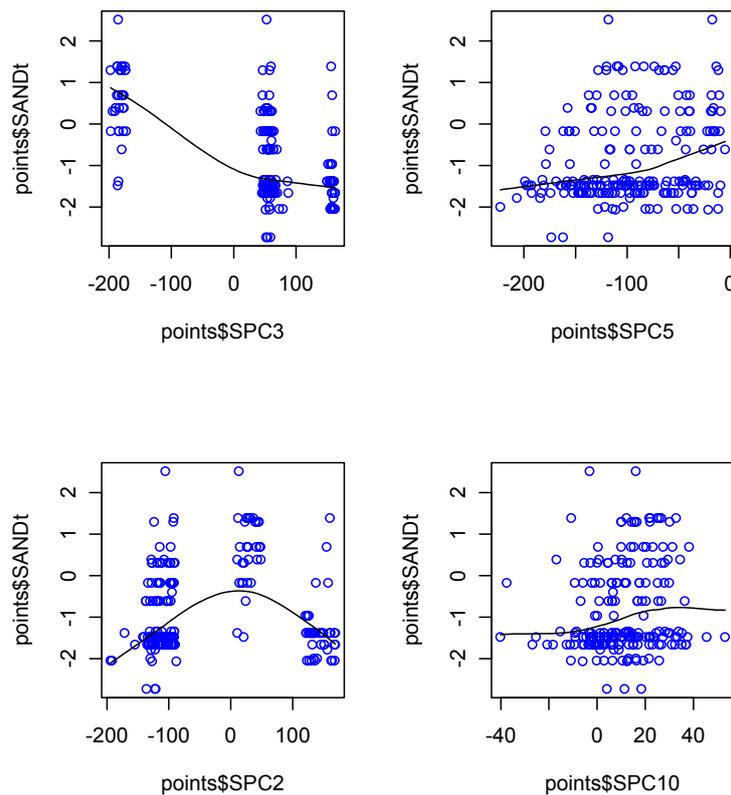
Fig. 4.13: Individual correlation plots fitted using a local polynomial. Such plots are useful to determine if linear or non-linear regression models are appropriate for fitting.

4.3.3   Multinomial logistic regression

We can fit the multinomial logistic regression model (§2.3) for variable SOILTYPE using the nnet package:

```
library(nnet)
```

```
soiltype.mnr <- multinom(SOILTYPE~SPC1+SPC2+SPC3+SPC4+SPC5+SPC6+SPC7+
SPC8+SPC9+SPC10+SPC11+SPC12, points)
```

this will fit regression coefficients for each soil type (12 predictors times 13 soil types) by using the single-hidden-layer neural network fitting (Venables and Ripley, 2002). This produces:

```
> summary(soiltype.mnr)

Call:
multinom(formula = SOILTYPE ~ SPC1 + SPC2 + SPC3 + SPC4 + SPC5 +
    SPC6 + SPC7 + SPC8 + SPC9 + SPC10 + SPC11 + SPC12, data = points)

Coefficients:
    (Intercept)          SPC1          SPC2 ...
B   -0.31896709  0.079545183   0.130273847 ...
D   -0.03515756  0.011957201   0.045788656 ...
G   -0.01827031  0.155334355   0.186910746 ...
```

```
Ha -0.02908207 -0.008472696  0.002753135 ...
K    0.25693393  0.034774367  0.057760249 ...
L    0.43391738 -0.001124392  0.008077742 ...
N   -0.01964908 -0.007922919  0.014177197 ...
NA  -0.02320083 -0.018915250 -0.006591750 ...
Q   -0.15163013 -0.009846705  0.012511021 ...
R   -0.01137301 -0.025438786  0.005745228 ...
S    0.06100672 -0.007516569  0.012431058 ...
Z   -0.10011658 -0.017814427  0.027012844 ...

Residual Deviance: 586.4614
AIC: 898.4614
```

which does not say much about which classes have been fitted less successfully. We only get an overall AIC[14] of 898.5, which does not say if the model is significant or not. Once we fitted the model, we can predict the values for SOILTYPE at all grid nodes using:

```
SOILTYPE.reg = predict(soiltype.mnr, newdata=SPC)

# Copy the predicted values to a spatial grid dataframe:

predictors$SOILTYPE.reg = SOILTYPE.reg
spplot(predictors["SOILTYPE.reg"], col.regions=bpy.colors(),
   scales=list(draw=TRUE), sp.layout = list("sp.points", pch=19,
   col="black", fill=T, points))
```

The resulting map (Fig. 4.14) indicates that the distribution of soil types is controlled by DEM parameters mainly (TWI and DEM25). Let us compare the frequencies of the classes at observation points and in the output map:

```
> summary(points$SOILTYPE)
 A  B  D  G Ha  K  L  N NA  Q  R  S  Z
10 60 15  6  1 15 78  3 24 33  3 35 17

> summary(SOILTYPE.reg)
    A     B     D     G    Ha     K     L     N    NA     Q     R     S     Z
 2730 24330   591  3186   612  2095 45888  1963 27027 19481  9640  3115 19342
```

which clearly shows that some classes, especially D and S, seem to be under-represented (smoothed out) by the multinomial logistic regression. Still, the predictors seems to be significant as many predicted classes show clear matching with the patterns of the predictors.

Now we can compare how well have a specific classes has been fitted versus the original observations:

```
points$SOILTYPE.L=ifelse(points$SOILTYPE=="L", 1, 0)
plot(points[sel,]$SOILTYPE.L, soiltype.mnr$fitted.values[,"L"], asp=1)
```

If you would explore each specific class, you will note that, in most cases, the fitted values always tend to smooth the actual observed values (lower mean and standard deviation), which can be easily seen by looking at the summary statistics. To export the result of spatial prediction to a GIS you will need to convert factors to numeric values:

---

[14]Akaike Information Criterion — the smaller the value, the better the model, but there is no statistical significance associated with this number as with adjusted R-square.
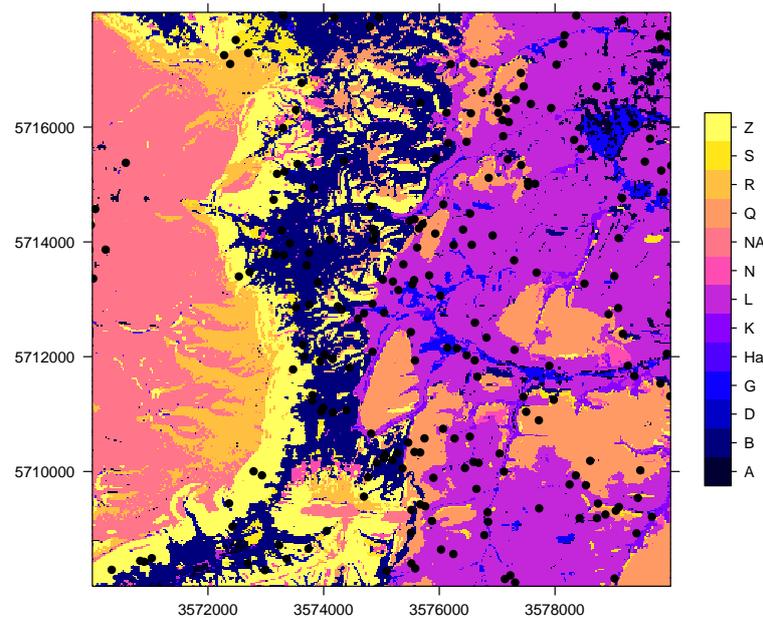
Fig. 4.14: A categorical variable (SOILTYPE) predicted using the multinomial logistic regression as implemented in the nnet package.

```
predictors$SOILTYPE.regn=as.numeric(predictors$SOILTYPE.reg)
writeGDAL(predictors["SOILTYPE.regn"], "ilwis/SOILTYPE_regn.mpr", "ILWIS")
```

Another possibility is to fit each class separately using the General Linear Models functionality of R. In this case, we only need to specify the link function (link=logit) and R will fit a logistic regression model using the iteratively reweighted least squares:

```
soiltype.L.glm = glm(SOILTYPE.L~SPC1+SPC2+SPC3+SPC4+SPC5+SPC6+SPC7+SPC8+SPC9
+SPC10+SPC11+SPC12, binomial(link=logit), points)
```

which gives a single logistic regression model:

```
> summary(soiltype.L.glm)

Call:
glm(formula = SOILTYPE.L ~ SPC1 + SPC2 + SPC3 + SPC4 + SPC5 +
    SPC6 + SPC7 + SPC8 + SPC9 + SPC10 + SPC11 + SPC12,
    family = binomial(link = logit), data = points)

Deviance Residuals:
    Min       1Q    Median       3Q       Max
-1.5247  -0.8600  -0.3381   0.9268    2.3375

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.100e+02  4.207e+02  -1.450    0.147
SPC1         4.222e+00  2.903e+00   1.454    0.146
SPC2         5.383e+00  3.711e+00   1.450    0.147
SPC3        -1.812e+00  1.250e+00  -1.450    0.147
SPC4        -3.337e+00  2.300e+00  -1.451    0.147
SPC5         5.164e-01  3.629e-01   1.423    0.155
```

```
SPC6          -1.332e+00  9.187e-01  -1.450    0.147
SPC7           1.999e+00  1.371e+00   1.458    0.145
SPC8          -2.416e-01  1.719e-01  -1.406    0.160
SPC9           1.261e+00  8.741e-01   1.442    0.149
SPC10         -1.294e-01  8.026e-02  -1.612    0.107
SPC11         -1.293e-01  1.033e-01  -1.252    0.211
SPC12          5.124e-04  3.336e-02   0.015    0.988


(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 257.08  on 221  degrees of freedom
Residual deviance: 205.03  on 209  degrees of freedom
  (78 observations deleted due to missingness)
AIC: 231.03

Number of Fisher Scoring iterations: 12
```
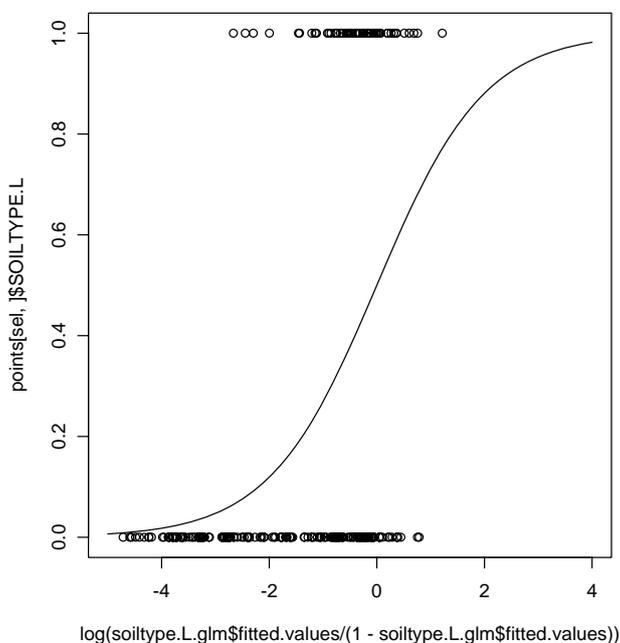

Fig. 4.15: Soil type "L" fitted using the logistic regression. The line indicates the conversion function (`link=logit`).

A result of fitting this soil type can be see in Fig. 4.15. Note that the fitted probability values will rarely exceed value of 0.8, which means that the model has much less problems to detect where the soil type does not appear rather to detect the 0 values. We can proceed with other soil types and then compare their AIC's. This would gives us much better insight into the success of data fitting and will allow us to detect problematic observations and classes.

4.4   Variogram modelling

4.4.1   Interpretation of the variograms

Now that we have fitted a regression model, we can proceed with fitting the variograms for the regression residuals. This can be done using the gstat package and its automated variogram fitting options. Let us first see how does the variogram of the original variable look like:

```
library(gstat)
plot(variogram(SAND~1, points), plot.nu=T, pch="+")
```

which will produce the plot in Fig. 1.7c. We can further on fit the variogram by providing an initial variogram:

```
sand.v = variogram(SAND~1, points)
sand.ovgm = fit.variogram(sand.v, vgm(nugget=25, model="Exp", range=455,
sill=423))
```

Note that we have determined the initial variogram by using the mean distance to nearest neighbour to estimate the range, measurement error to estimate the nugget and global variance to estimate the sill parameter (less nugget). The fitted R model has the following structure:

```
> str(sand.ovgm)
Classes 'variogramModel' and 'data.frame':     2 obs. of  9 variables:
 $ model: Factor w/ 17 levels "Nug","Exp","Sph",..: 1 2
 $ psill: num   26 440
 $ range: num    0 478
 $ kappa: num  0 0.5
 $ ang1 : num  0 0
 $ ang2 : num  0 0
 $ ang3 : num  0 0
 $ anis1: num  1 1
 $ anis2: num  1 1
 - attr(*, "singular")= logi FALSE
 - attr(*, "SSErr")= num 2.61
```

where the anisotropy parameters are set to 1, which means that the model is isotropic. Both the experimental variogram and the fitted model can be visualized using (Fig. 1.7d):

```
plot(sand.v, sand.ovgm, plot.nu=F)
```

From this plot we can see that our initial estimate of the variogram was, in fact, quite accurate. The fitted nugget parameter is 26, sill parameter is 448 and the sill is at 440. We will actually work with the logit-transformed value of the target variable, which will give us a nugget parameter of 0.084, a sill parameter of 1.012 and a range parameter of 478 m. Recall from §4.2.1 that the global variance for the logit-transformed values of the target variable (SANDt) is 0.99.

4.4.2   Variograms of residuals

Next we need to estimate variograms of the residuals. To achieve this, we extend the trend model using the (selected) SPCs:

```
sand.rv = variogram(SANDt~SPC2+SPC3+SPC4+SPC5+SPC10+SPC11, points[sel,])
```

we have masked out[15] the points which are outside the study area because `gstat` can not calculate variograms with `NA`s. The same command can be written more elegantly as:

```
sand.rev = variogram(fsand$call$formula, points[sel,])
```

which means the we will automatically pass the output formula from the stepwise regression (`fsand$call$formula`), so that we do not need to type it manually. Now we fit the variogram of residuals by using the same values used for the standard initial variogram, but we set the sill variance at half[16]:

```
sand.rvgm = fit.variogram(sand.rev, vgm(nugget=0.08, model="Exp",
    range=228, sill=0.5))
```

which gives the following parameters: C_0=0.080, C_1=0.491, R=257 m. Again, our estimate of the initial variogram has been quite accurate. The fitted variograms of the target variable and residuals are shown in Fig. 4.16.
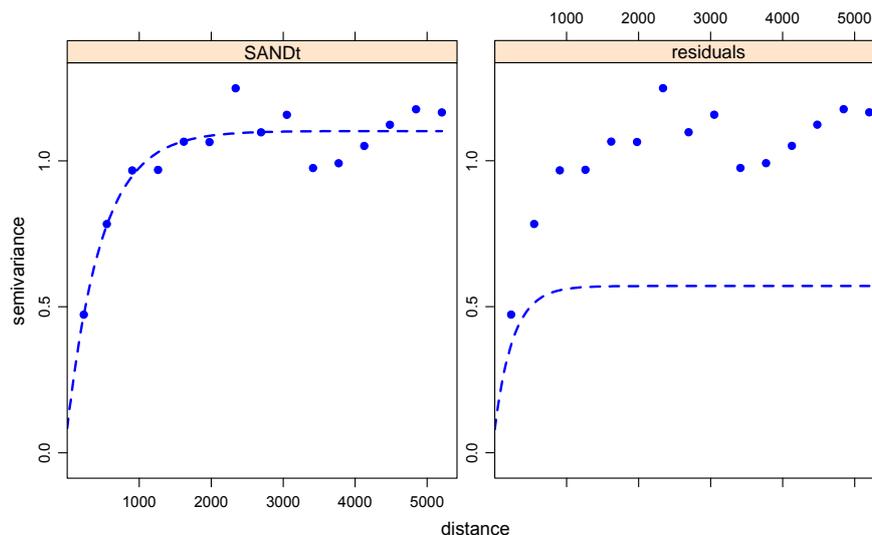
Fig. 4.16: Fitted variograms for target variable `SANDt` (left) and residuals (right).

You might wonder why the variograms of the residuals and of the target variable are different? In theory (Eq.1.1.1), the deterministic and the stochastic part of variation are independent and add to each other. In practice, they are intermixed. If the predictors are generally smooth, and if they are correlated with our target variables, then the variogram will reflect the spatial structure of the predictors. Hence, the variogram of the target variable will not only show spatial-autocorrelation of the stochastic part of variation, but also of the deterministic part. You can see that indeed all predictors are in fact very smooth by using:

---

[15]By using a subset of data: `sel=!is.na(points$SPC1)`.

[16]This assumes that the regression model will only scale the variogram of the residuals — the nugget variation and the range of spatial auto-correlation should remain the same.

```
SPC1.v = variogram(SPC1~1, points[sel,])
SPC1.ovgm = fit.variogram(SPC1.v, vgm(nugget=0, model="Exp", range=500,
sill=1000))
plot(SPC1.v, SPC1.ovgm, pch="+", plot.nu=T)
```

We continue fitting the variograms also for other texture fractions and get similar models: C_0=0.145, C_1=0.240, R=350 m for SILTt and C_0=0.150, C_1=0.160, R=304 m for CLAYt. In this case, we were relatively lucky with fitting the variogram automatically. In other situations, you will notice that non-linear least square fitting is only guaranteed to work when a good initial values are set. In any case, visual validation of the model fit is often recommended, even though the fitting can be very successful (Pebesma, 2004).

The variograms for residuals of the logistic regression model for soil types can be derived using:

```
plot(variogram(soiltype.L.glm$residuals~1, soiltype.L.glm$data[sel,]))
```

this show a relatively high nugget variation, mainly because of one extremely high value (ID=16). Try to produce similar variograms also for other soil types and plot them next to each other.

4.5   Predictions and simulations

Now that we have fitted the parameters of the regression model (significant predictors and their regression coefficients) and of the variograms (nugget, sill and range), we can use this model to derive predictions at all locations. Regression-kriging can be run in gstat by using:

```
SAND.rk = krige(fsand$call$formula, points[sel,], SPC, sand.rvgm)
```

where fsand$call$formula is the regression model fitted using the step-wise regression, points is the input point dataset to be interpolated, SPC is the list of 12 rasters, and sand.rvgm is the fitted variogram model of residuals. The computation might take even several minutes, because the system needs to invert large matrices (222 points) and then make predictions at 160,000 grid nodes.

Let us look at the structure of the output data set:

```
>str(SAND.rk)
Formal class 'SpatialPixelsDataFrame' [package "sp"] with 7 slots
  ..@ data        :'data.frame': 160000 obs. of  2 variables:
  .. ..$ var1.pred: num [1:160000] -1.078 -0.917 -0.944 -1.006 -1.400 ...
  .. ..$ var1.var : num [1:160000] 0.597 0.603 0.598 0.593 0.607 ...
  ..@ coords.nrs : num(0)
  ..@ grid        :Formal class 'GridTopology' [package "sp"] with 3 slots
  .. .. ..@ cellcentre.offset: Named num [1:2] 3570013 5708013
  .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
  .. .. ..@ cellsize         : Named num [1:2] 25 25
  .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
  .. .. ..@ cells.dim        : Named int [1:2] 400 400
  .. .. .. ..- attr(*, "names")= chr [1:2] "x" "y"
  ..@ grid.index : int [1:160000] 1 2 3 4 5 6 7 8 9 10 ...
  ..@ coords     : num [1:160000, 1:2] 3570013 3570038 3570063 3570088 3570113 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : NULL
```

```
.. .. ..$ : chr [1:2] "x" "y"
..@ bbox       : num [1:2, 1:2] 3570000 5708000 3580000 5718000
.. ..- attr(*, "dimnames")=List of 2
.. .. ..$ : chr [1:2] "x" "y"
.. .. ..$ : chr [1:2] "min" "max"
..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
.. .. ..@ projargs: chr " +init=epsg:31467
```
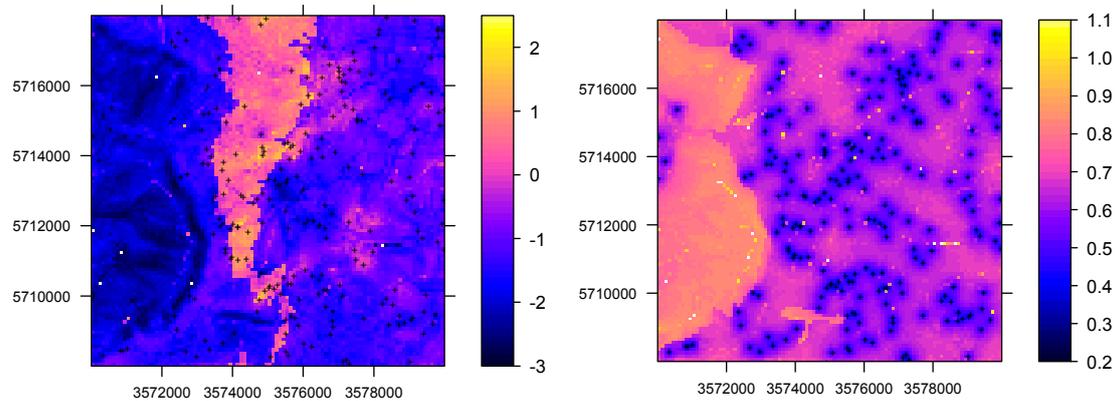

Fig. 4.17: `SANDt` interpolated using regression-kriging (at 100 m grid): predictions (left) and the prediction variance (right).

This means that `krige` produced two maps: (1) `"var1.pred"` — predictions and (2) `"var1.var"` — prediction variance or estimated error of mapping a variable. This data frame has exactly the same grid definition as the input raster maps, however, the class has changed to `SpatialPixelsDataFrame` (point support). We can plot the two maps next to each other by using the spplot command (Fig. 4.17):

```
sand.rkpred1.plt = spplot(SAND.rk["var1.pred"], col.regions=bpy.colors(),
    scales=list(draw=TRUE, cex=0.7), sp.layout = list("sp.points", pch="+",
    col="black", fill=T, points))
sand.rkvar.plt = spplot(SAND.rk["var1.var"], col.regions=bpy.colors(),
    scales=list(draw=TRUE, cex=0.7), at = seq(0.1,0.7,0.02), sp.layout =
    list("sp.points", pch="+", col="black", fill=T, points))

print(sand.rkpred1.plt, split=c(1,1,2,1), more=TRUE)
print(sand.rkvar.plt, split=c(2,1,2,1), more=FALSE)
```

where argument `col.regions` defines the legend, `scales` will plot the coordinates and `sp.layout` defines the overlays. You can now compare these results with the results of spatial prediction using more trivial spatial prediction techniques shown in Fig. 1.11. Note that the prediction variance map (Fig. 4.17, right) indicates the areas of extrapolation in both geographical and feature spaces. Recall from §4.7.2 that our sampling design is under-representing some areas (especially `Z1`). The map in Fig. 4.17 confirms that the biggest extrapolation is exactly in the areas that have been under-sampled: geological zone `Z1` and areas of high hydrological potential (stream bottoms).

In addition to predictions, we can opt to produce simulations using the same regression-kriging model by adding an additional argument:

```
SAND.rksim = krige(fsand$call$formula, points[sel,], SPC,
            sand.rvgm, nsim = 6)
```
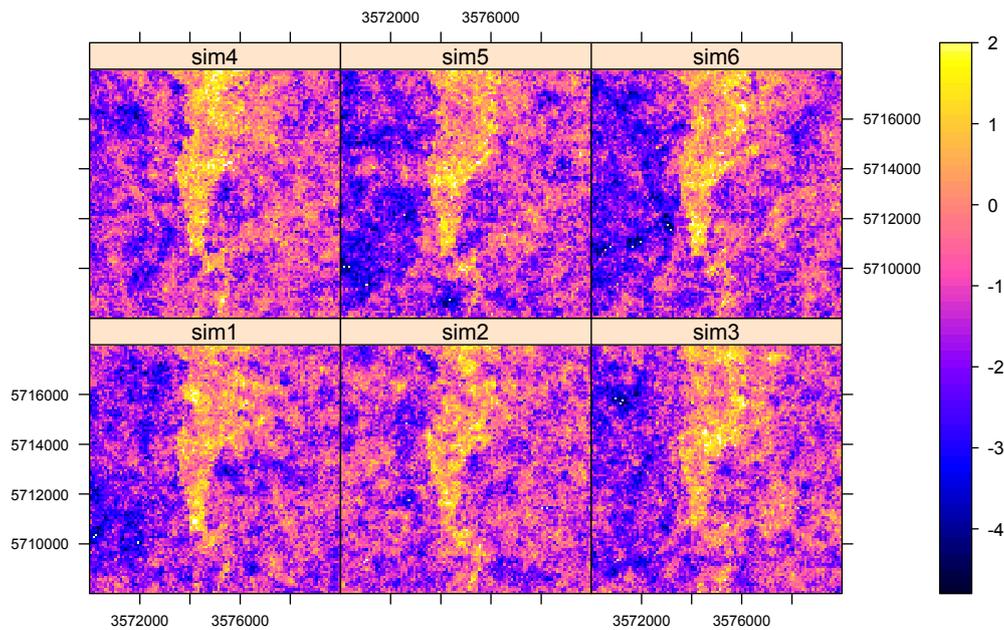
Fig. 4.18: SANDt: six equiprobable realizations of the same regression-kriging model.

which will produce 6 equiprobable realizations of this model using the Sequential Gaussian Simulation algorithm (see §2.4 for more details). Note that this algorithm can be computationally extensive for large point and raster datasets, so that it is often advisable to limit the search size[17] e.g. using nmax=100. The output of computation is now a grid data set with five bands (Fig. 4.18). Such simulations are very useful to visually explore uncertainty of mapping an environmental variable and can be used to assess the propagated uncertainty of using such variables in decision making.

Visual exploration of such simulations can also be very useful to judge about the importance of the predictors. For example, by looking closely at the Fig. 4.18, we can see that the patterns of SAND are indeed controlled by unit Z3 and map of TWI, while the other predictors seem to be less significant.

Before we can evaluate the results of spatial prediction, we need to back-transform the predictions to the original scale by:

```
SAND.rk$pred = exp(SAND.rk$var1.pred)/(1+exp(SAND.rk$var1.pred))*100
```

which will create an additional spatial layer in the SAND.rk grid data frame. We can produce the same predictions for both SILT and CLAY and then export the resulting maps to ILWIS.

To map a specific soil type class, you can first predict the deterministic part of variation using the logistic regression model, then interpolate the residuals using the variogram model fitted previously:

```
SOILTYPE.Lt.reg = predict(soiltype.L.glm, newdata=SPC)
predictors$SOILTYPE.Lt.reg = SOILTYPE.Lt.reg

SOILTYPE.Lt.resok = krige(soiltype.L.glm$residuals~1, soiltype.L.glm$data[sel,],
    SPC, soiltype.L.rvgm)
```

---

[17]Recall from §2.2 that this is not really a valid thing to do.
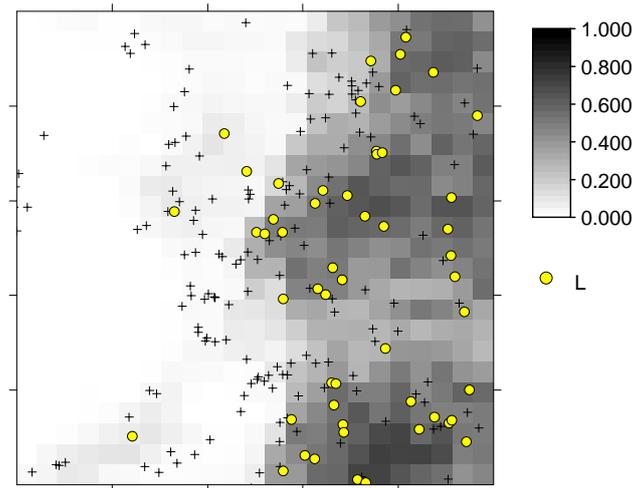
Fig. 4.19: Soil type "L" interpolated using regression-kriging. Compare with Fig. 4.14.

Once you estimated both the deterministic part and the residuals, you can sum them and back transform the logit values to the original 0–1 scale:

```
predictors$SOILTYPE.L.rk = exp(predictors$SOILTYPE.Lt.reg +
        SOILTYPE.Lt.resok$var1.pred)/(1+exp(predictors$SOILTYPE.Lt.reg +
        SOILTYPE.Lt.resok$var1.pred))

spplot(predictors["SOILTYPE.L.rk"], col.regions=bpy.colors(), at = seq(0,1,0.02),
    sp.layout = list("sp.points", pch="+", cex=1.2, col="white", points))
```

The final map showing predicted odds of observing soil type "L" in the area is shown in Fig. 4.19. Note that now all the values are within the 0–1 range and the distribution of this soil class seems to closely follow specific landscape positions.

4.6   Assessing the quality of predictions

RK variance is the statistical estimate of the model uncertainty. Note that the '*true*' prediction power can only be assessed by using the independent (control) data set. The prediction error is therefore often referred to as the *precision of prediction*. The true quality of a map can be best assessed by comparing estimated values ($\hat{z}(\mathbf{s}_j)$) with actual observations at validation points ($z^*(\mathbf{s}_j)$). Commonly, two measures are most relevant here — (1) the mean prediction error (ME):

$$ME = \frac{1}{l} \cdot \sum_{j=1}^{l} [\hat{z}(\mathbf{s}_j) - z^*(\mathbf{s}_j)]; \qquad E\{ME\} = 0 \qquad (4.6.1)$$

and (2) the root mean square prediction error ($RMSE$):

$$RMSE = \sqrt{\frac{1}{l} \cdot \sum_{j=1}^{l} [\hat{z}(\mathbf{s}_j) - z^*(\mathbf{s}_j)]^2}; \qquad E\{RMSE\} = \sigma(\mathbf{h} = 0) \qquad (4.6.2)$$

where l is the number of validation points. We can also standardize the errors based on the prediction variance estimated by the spatial prediction model:

$$RMNSE = \sqrt{\frac{1}{l} \cdot \sum_{j=1}^{l} \left[\frac{\hat{z}(\mathbf{s}_j) - z^*(\mathbf{s}_j)}{\hat{\sigma}_j} \right]^2}; \qquad E\{RMNSE\} = 1 \qquad (4.6.3)$$

In order to compare accuracy of prediction between variables of different type, the *RMSE* can also be normalised by the total variation:

$$RMSE_r = \frac{RMSE}{s_z} \qquad (4.6.4)$$

which will show how much of the global variation budget has been explained by the model. As a rule of thumb, a value of $RMSE_r$ that is close to 40% means a fairly satisfactory accuracy of prediction (R-square=85%). Otherwise, if $RMSE_r > 71\%$, this means that the model accounted for less than 50% of variability at the validation points. Note also that *ME*, *RMSE* and *RMNSE* estimated at validation points are also only a sample from a population of values — if the validation points are poorly sampled, so will our estimate of the map quality be poor.

To assess the accuracy of predicting the categorical variables we can use the **kappa statistics**, which is a common measure of classification accuracy (Congalton and Green, 1999; Foody, 2004). Kappa statistics measures the difference between the actual agreement between the predictions and ground truth and the agreement that could be expected by chance. In most remote sensing-based mapping projects, a kappa larger than 85% is considered to be a satisfactory result (Foody, 2004). The kappa is only a measure of the overall mapping accuracy. Specific classes were analyzed by examining the percentage of correctly classified pixels per each class:

$$P_c = \frac{\sum_{j=1}^{m} \left(\hat{C}(s_j) = C(s_j) \right)}{m} \qquad (4.6.5)$$

where P_c is the percentage of correctly classified pixels, $\hat{C}(s_j)$ is the estimated class at validation locations (s_j) and m is total number of observations of class c at validation points.

We can now compare the quality of predictions for mapping soil texture fractions. First we need to import the table with locations of validation points and the measured values of target variables. For this, we will use the large set of points (l=2937):

```
pointsal = read.dbf("pointsal.dbf")
coordinates(pointsal)=~x+y
proj4string(pointsal) = CRS("+init=epsg:31467")
```

then, we overlay the validation points and the produced maps:

```
pointsalSANDrk.ov = overlay(SAND.rk, pointsal)
```

and we can derive *ME*, *RMSE* and *RMNSE*:

```
E.SAND.rk=pointsal$SAND - pointsalSANDrk.ov$pred
EN.SAND.rk=(pointsal$SANDt - pointsalSANDrk.ov$var1.pred)/
          sqrt(pointsalSANDrk.ov$var1.var)
mean(na.omit(E.SAND.rk))
sqrt(mean(na.omit(E.SAND.rk)^2))
sqrt(mean(na.omit(EN.SAND.rk)^2))
```

Now it might be interesting to plot the predicted values versus the measured values at control points and derive the correlation coefficient. As we can see in Fig. 4.20, the models is somewhat biased in predicting lower and higher values. The correlation coefficient shows that the predictions and measured values at validation points are significantly correlated (r=0.76). What surprises us more is that the prediction variance does not seems to be correlated with the true errors (Fig. 4.20, right). We can see that the prediction errors will be in average higher in areas where the model is less certain, still the two variables does not seem to be significantly correlated. Although this allows you to criticise the usability of the prediction variance, you should not that also these validation points have been sampled using the same sampling strategy. If we have used more independent validation points, we would have probably detected better match between the estimated and true prediction variance.
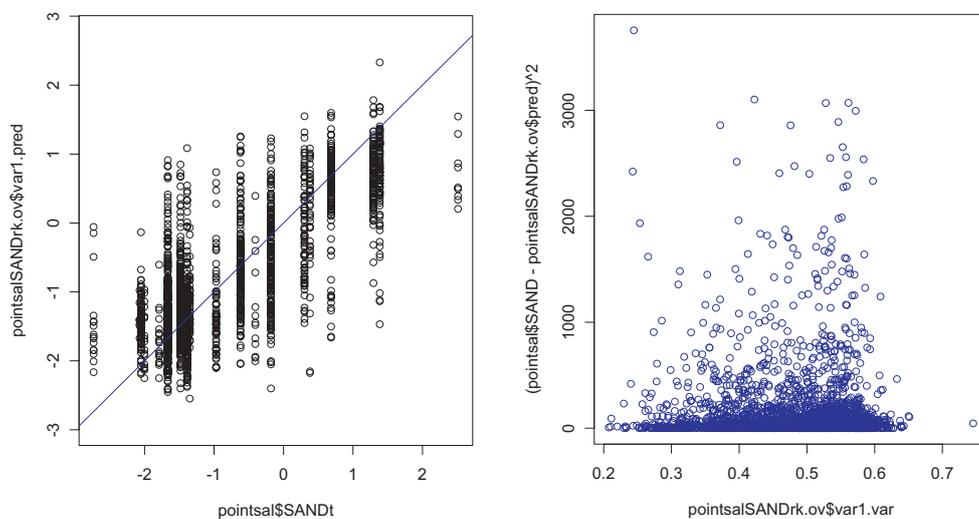
Fig. 4.20: Predicted versus measured values (left); estimated prediction error versus the measured/true error (right).
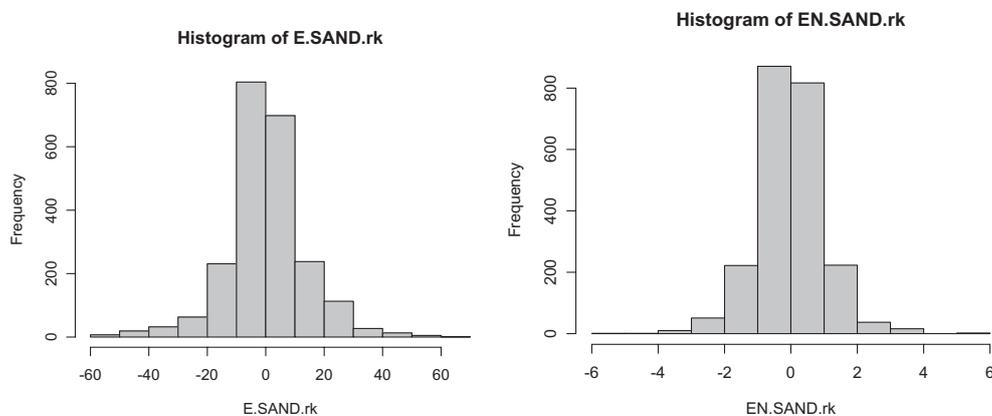
Fig. 4.21: Original and normalized errors for predicting `SAND` using RK technique (l=2937).

Another useful thing to do is to examine the histogram of errors at validation points and compare the errors estimated by the model (prediction variance) and the true mapping error at validation points. This can help us to detect 'unusual' locations where the errors are much higher than at other locations. According to the Chebyshev's Inequality theorem, proportion of normalized errors that exceed value of 3 should not be higher than 1/9. In our case study (Fig. 4.21), histograms of *RMNSE* for RK show that these maps satisfy the Chebyshev's Inequality theorem.

We can repeat this procedure also for the OK predictions. After we have calculated the validation measures, we might test the difference between the distributions of the prediction errors for two methods by using:

```
> t.test(E.SAND.rk, E.SAND.ok)


        Welch Two Sample t-test

data:  E.SAND.rk and E.SAND.ok
t = 1.7762, df = 4408.031, p-value = 0.07576
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.07990234  1.62030858
sample estimates:
  mean of x    mean of y
 0.05961023 -0.71059289
```

which shows that RK performs better, but this difference is not statistically significant at the 0.05 probability level. Continue the same comparison for both `SILT` and `CLAY` and you will see that RK significantly improves predictions of `CLAY`, marginally improves predictions of `SAND` and does not make better prediction of `SILT`. Going back to the properties of our input dataset, we see that the areas of high `CLAY` content have been under-sampled (see further Fig. 4.26), hence a plain geostatistical spatial prediction technique such as OK will perform poorer.

Table 4.1: Comparison of performance of regression-kriging versus ordinary kriging using 2937 validation points (`pointsal`).

var	Sampled (n=222)			Regression-kriging			Ordinary kriging		
	mean	s.d.	$\sqrt{C_0}$	*ME*	*RMSE*	*RMNSE*	*ME*	*RMSE*	*RMNSE*
`SAND`	30.2	20.5	5.0	0.1	13.5	1.00	-0.7	15.6	1.00
`SILT`	48.2	18.4	2.1	0.5	12.9	0.99	0.4	14.3	1.15
`CLAY`	21.6	11.8	6.6	1.2	7.9	1.03	-0.1	8.3	1.05

A summary comparison of the performance of regression-kriging versus ordinary kriging can be seen in Table 4.1. Note that RK outperforms OK in all cases[18], however, the difference is not always significant. Also note that there is still a significant amount of variation that could be mapped until we would reach the measurement error[19]. The question remains if we could achieve this by using higher quality predictors or we simply need to collect more point data.

We proceed with comparing the accuracy of predictions for mapping variable `SOILTYPE`. For this, we need to install a package for discriminant analysis (mda). First we need to overlay the predicted classes and the validation dataset:

---

[18]Note that RK is making somewhat biased estimation of `CLAY` with *ME*=1.2.

[19]Recall that the precision of measuring the soil texture by hand is about ±5–10%, which corresponds to the fitted nugget ($\sqrt{C_0}$).

```
pointsalSOILTYPE.ov = overlay(predictors, pointsal)
```

then, we derive the confusion matrix for the two factors (true vs predicted):

```
> library(mda)
> confusion(pointsalSOILTYPE.ov$SOILTYPE.reg, pointsal$SOILTYPE)
      true
object  A   B  D  G  K   L N NA  Q R   S  Z Hw
    A  10   4  0  4  5  10 0  9  1 0   2  0  1
    B   3 278 56  3 19  71 9 19 61 1  62 56  0
    D   0   2  3  0  0   0 0  0  0 0   0  1  0
    G   8   4  0 19 10   8 0 10  1 0   6  0  0
    K   0   3  0  1  5   2 0  1  1 0   2  0  0
    L  13 131  6 23 74 287 3 13 79 0 176  2  0
    N   0   4  1  0  0   0 0  0  1 0   1  0  0
    NA  0   1  0  0  1   2 0 77  1 0   0  0  0
    Q   0  66  2  6  0  21 1  3 86 1  77  1  0
    R   0   5  2  0  0   0 0  0  0 9   1  2  0
    S   0  17 10  0  2   7 0  2  8 1  12 10  0
    Z   0  39 75  0  0   2 0  1  7 6  13 77  0
    Ha  3   3  0  2  1   0 0  0  0 0   3  0  0
attr(,"error")
[1] 0.6143878
attr(,"mismatch")
[1] 0.005775211
```
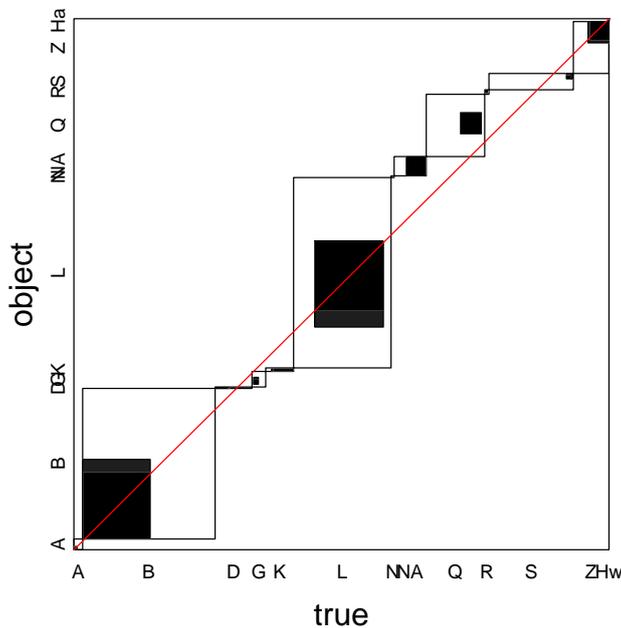

Fig. 4.22: Agreement plot for mapping variable `SOILTYPE` using multinomial logistic regression. The size of the white rectangles indicate the portion of a class and the black rectangles indicate the success of prediction.

This shows that several classes are completely poorly predicted (e.g. `"D"`, `"K"` and `"S"`). Another useful package for analysis and visualization of categorical data is the

Visualizing Categorical Data (vcd) package. This can be used to visually explore the agreement between the predicted and observed classes:

```
library(vcd)
agreementplot(confusion(pointsalSOILTYPE.ov$SOILTYPE.reg,
pointsal$SOILTYPE))
```

As you can see from Fig. 4.22 the most successfully mapped classes are "B", "L", "R" and "Z". We finally derive the kappa statistics using:

```
> Kappa(confusion(pointsalSOILTYPE.ov$SOILTYPE.reg, pointsal$SOILTYPE))
              value        ASE
Unweighted 0.2629102 0.01225020
Weighted   0.2344687 0.02409146
```

which shows that the predictions are successful at only 25% of the validation points. This might not be that bad, however, it shows that the map shown is Fig. 4.14 is far from a quality product (although it visually seems to be fine). The kappa is a rather strict measure of quality. We would obtain a more promising result if we would merge some of the classes, or if we would instead consider calculating a weighted kappa.

4.7 Comparison of predictions using various inputs

4.7.1 Importance of the cell size

The focus of this exercise is to examine how much the cell size of the input maps has influence on the accuracy of the final predictions. The cell size can be closely related to the level of detail or spatial precision of a map, which, in cartography, is often related to the concept of **scale**. Enlarging the cell size leads to **aggregation** or upscaling; decreasing the cell size leads to **disaggregation** or downscaling. As the grid becomes coarser, the overall information content in the map will progressively decrease, and vice versa (McBratney, 1998; Kuo et al., 1999; Stein et al., 2001). On the other hand, to do geostatistical mapping with very fine-resolution maps can be quite time-consuming. Ideally, we look for such cell size of our GIS that is '*good enough*' for mapping purposes.

Now we want to asses if the quality of predictions would decrease if we switch from the 25 m to 100 m cell size. To do this, we have to resample all input maps (e.g. in ILWIS) to the 100 m grid and then repeat all procedures as explained in §4.2.2 to derive the same predictors. Then, we can again derive the SPCs and import them to R and used them as predictions.

Fig. 4.23 shows a comparison of maps produced using the 100 m and 25 m predictors. Obviously, the 25 m maps show much finer detail than the 100 m maps, especially considering the hydrological features in the area. Note also that, at 100 m resolution, the model will produce much lower values for SAND in the areas that have been poorly sampled (Z1). Further comparison of errors at validation points shows that the RK at finer resolution is always more accurate than if the 100 m maps are used. The t-test shows that the two maps do not differ significantly in their accuracy for SAND (p=0.641) and SILT (p=0.608), however the 25 m maps can be used to map CLAY at significantly higher accuracy (p=0.044).
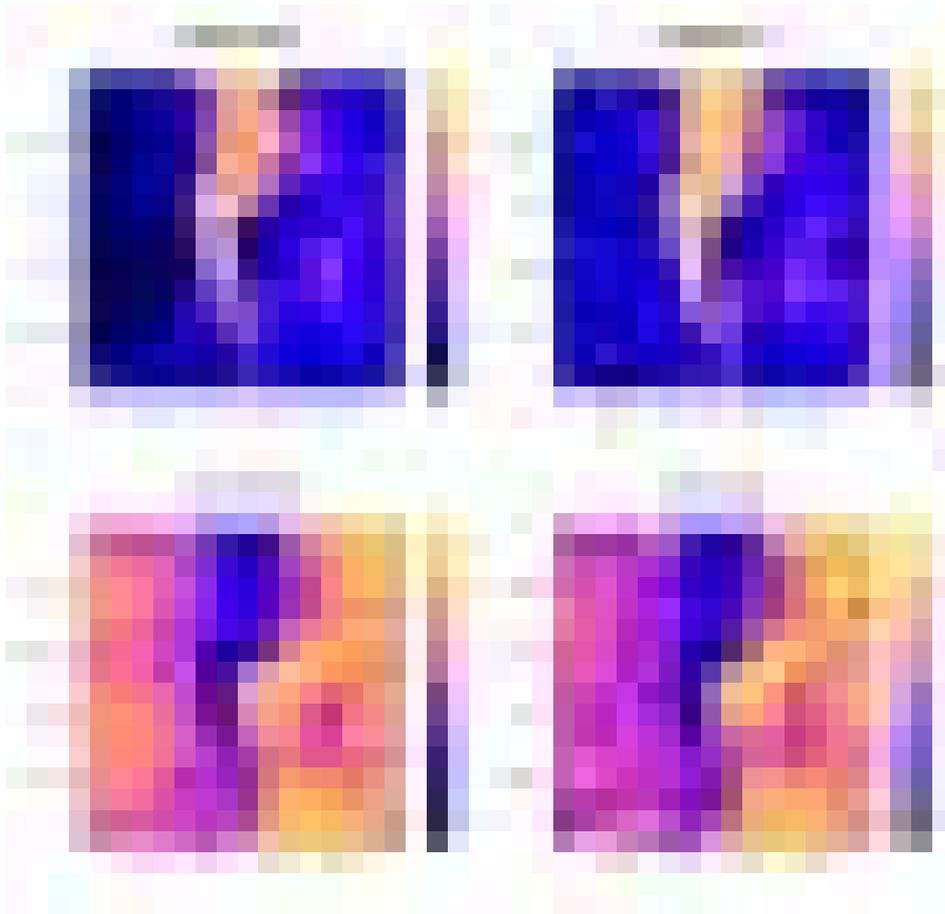
Fig. 4.23: Predictions of `SAND` and `SILT` produced using the 100 m and 25 m resolution maps.

4.7.2 Importance of the sampling intensity

In the next exercise, we want to see if the accuracy of the maps will increase significantly if we use the complete dataset (2251 points) versus the original sample that is about ten times smaller in size (222 points). We consider four combinations of prediction models: (1) OK using the large data set (`pointsal`), (2) OK using the small data set (`points`), (3) RK using the large data set (`pointsal`), and (4) RK using the small data set (`points`). We will compare the accuracy of mapping using the second `control` point data set.

We can already note that if we use the large point data set the adjusted R-square for the regression models will not change much: now the model explains 54% of variation for `SANDt`, 47% for `SILTt` and 52% for `CLAYt`. This proves that the success of regression modelling is not really dependent on the sampling density, but rather on how well are the points sampled in the feature space and how significant is the correlation. We continue with modelling the variograms for residuals and then run both RK and RK at grid resolution of 25 m. The final comparison can be seen in Fig. 4.24. Note that the two RK maps really do not differ so much visually. The OK maps, on the other hand, do differ both in the level of detail and in the mapping accuracy. The final comparison between the OK (2251) and RK (222) shows that investment in auxiliary predictors is indeed worth the effort — in all cases RK performs better. This indicates that future mapping projects will need to focus more on the quality of sampling and on quality of auxiliary environmental predictors, rather than on making more observations.
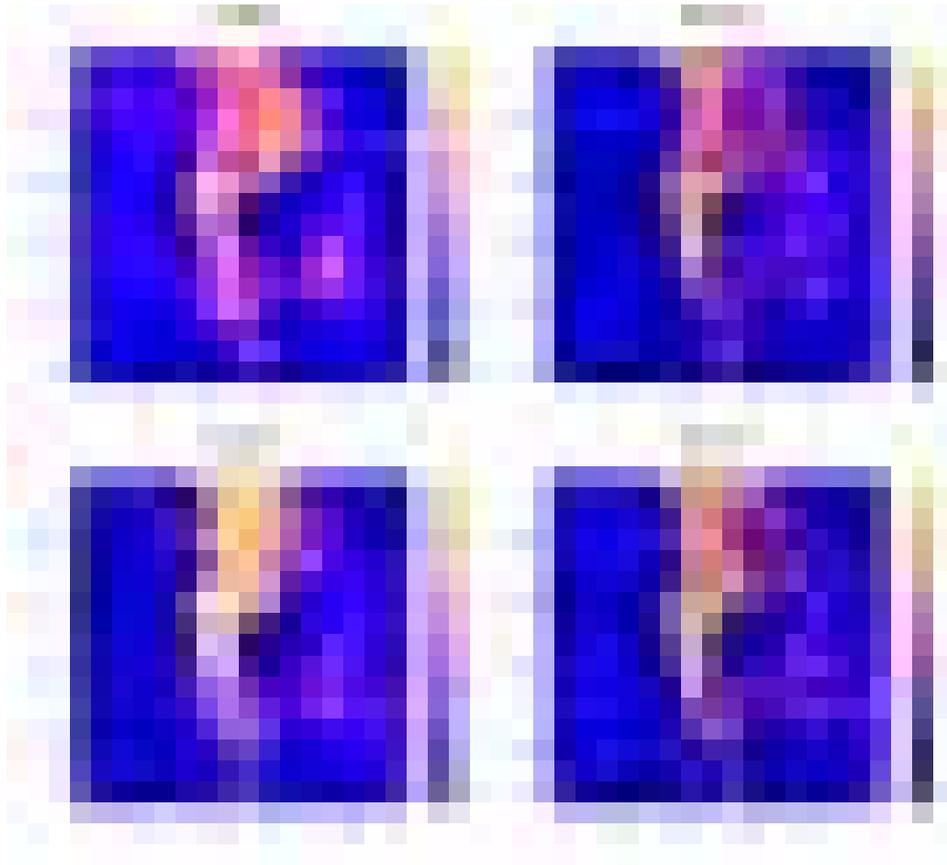
Fig. 4.24: `SAND` predicted using the large point data set (left) and 1/10 of the data set (right): ordinary kriging versus regression-kriging.

4.8   Visualization of the outputs

Although there are many options to visualize the derived maps in R (e.g. by using the `spplot` method), you might also want to export the maps to a GIS and then create more professional layouts. The following sections will provide instructions on how to prepare informative layouts in ILWIS and Google Earth.

4.8.1   Export to **ILWIS**

From R, you can at any time export produced maps to ILWIS by using the `writeGDAL` command:

```
writeGDAL(SAND.rk["pred"], "ilwis/SAND_rk.mpr", "ILWIS")
writeGDAL(SAND.rk["var1.var"], "ilwis/SAND_rk.mpr", "ILWIS")
```

Alternatively, you can export a map to the Arc/Info ASCII format by using:

```
write.asciigrid(SAND.rk, "SAND_rk.asc")
```

Once both the predictions and the prediction variance have been exported to ILWIS, they can be visualized jointly by using the visualization algorithm explained in §3.2.1. For each texture fraction, we need to have the predicted values, map of the prediction variance and know the estimated global variance. We can then run the ILWIS script `VIS_error`:

```
run 'C:\Program Files\Ilwis3\Scripts\VIS_error'  SAND_rk 5 90
SAND_rkvar 0.4 1.0 0.989 SAND_rkvis
run 'C:\Program Files\Ilwis3\Scripts\VIS_error'  SILT_rk 6 80
SILT_rkvar 0.4 1.0 0.682 SILT_rkvis
run 'C:\Program Files\Ilwis3\Scripts\VIS_error'  CLAY_rk 2 70
CLAY_rkvar 0.4 1.0 0.591 CLAY_rkvis
```
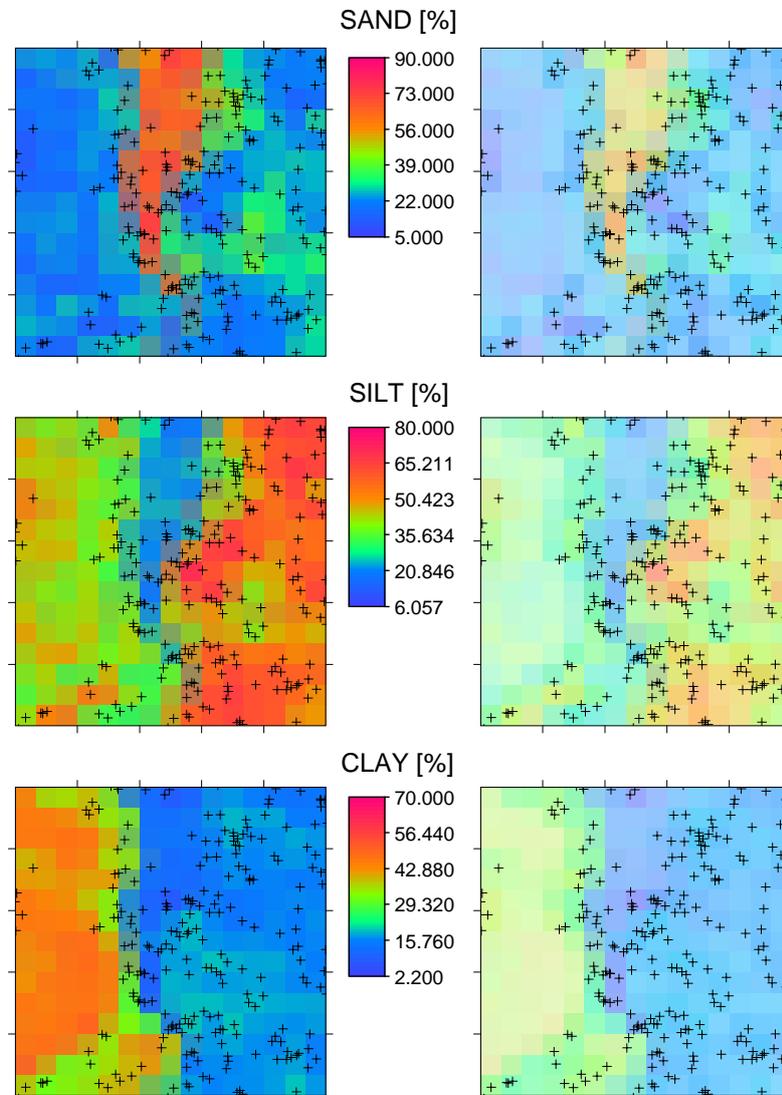

Fig. 4.25: Predicted values for `SAND`, `SILT` and `CLAY` visualized (left) without the prediction variance and (right) together with the prediction variance using the whitening technique.

The results of mapping the three texture fractions can be seen in Fig. 4.25. If you compare the original maps on the left side of Fig. 4.25 and the same maps visualized by including the mapping error, you can see that the predictions are in fact satisfactory: for all three texture fractions we can be relatively confident that the predictions are precise in >50% of the area. In the case of CLAY, the proportion of areas that are uncertain is much higher.

Another possibility to visualize the three texture fractions is to use a colour composite. Because they are compositional variables, the `SAND`, `SILT` and `CLAY` can be coded using the R, G, B bands. First, we need to convert the 0–100% values to the 0–255 image

domain in ILWIS. Then the three texture fractions can be combined using (Fig. 4.26):

```
TEXTrgb.mpr = MapColorComp24Linear(mlist(SAND_rkimg, SILT_rkimg,
CLAY_rkimg), 0:255, 0:255, 0:255)
```

We can also calculate the average normalized prediction error by using (Fig. 4.26):

```
TEXT_rkerror = ( sqrt(SAND_rkvar)/sqrt(0.989) + sqrt(SILT_rkvar)/sqrt(0.682)
  + sqrt(CLAY_rkvar)/sqrt(0.591) )/3
```
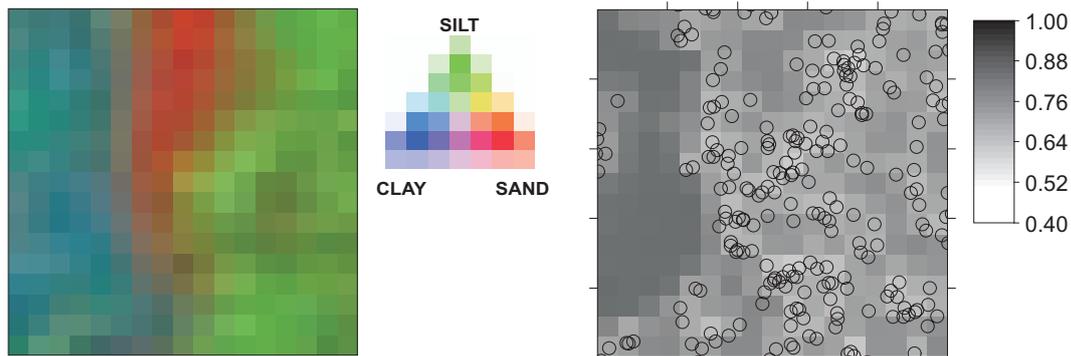

Fig. 4.26: Texture fractions visualized using a colour composite (left) and the average normalized prediction error for mapping texture fractions (right).

As expected, the highest prediction error is within the unit Z1. This average normalized prediction error can be used to allocate 100 additional points by using a weighted random sampling. First, convert the average normalized prediction error to penalty points[20]:

```
TEXTE=((TEXT_rkerror-0.4)*100)^2
```

Now import this map of standardized prediction error to R and use it as the weight map:

```
TEXTE = readGDAL("ilwis/TEXTE.mpr")

# Convert the raster map to an 'im' object:

TEXTEim = as.im(as.image.SpatialGridDataFrame(TEXTE))
new.points = rpoint(100, TEXTEim)
plot(new.points, pch="+")
```

this will produce a weighted random design with the inspection density proportional to the value of the standardized prediction error. Fig. 4.27 shows one realization of a weighted random sampling design that can be used to improve the precision of the map. A more sophisticated sampling optimization procedure can be found in Brus and Heuvelink (2007).

---

[20]If the normalized prediction variance is close to 0.4 than there is no need for sampling new points. See also Eq.1.3.19.
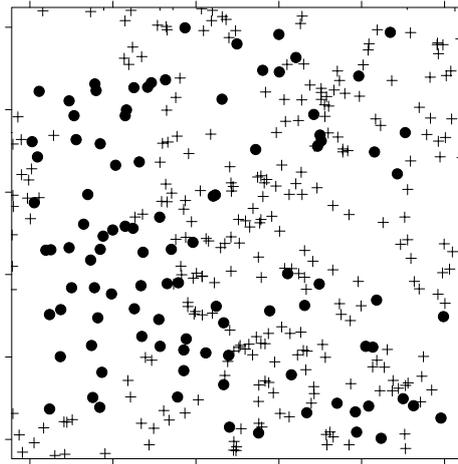
Fig. 4.27: Existing 222 points and the additional 100 points generated by using inspection density proportional to the value of the prediction variance.

4.8.2   Export to KML

Assuming that we have finished data analysis and generated all predictions, we can proceed with preparing the data for web-sharing. The most suitable option to share the results of a mapping project to a wider community is Google Earth. Before we can prepare KML files, we need to reproject/resample our raster maps to the LatLonWGS84 coordinate system. We have first estimated the definition of the new geographical grid by following the instructions in §3.5.2. This gives the following grid system for the Ebergötzen case study:

```
Lines=315
Columns=505
MinX=10.011111
MinY=51.504167
MaxX=10.151389
MaxY=51.591667
Cell=0.0002777778
```

Once we have created a georeference for the LatLonWGS84 coordinate system, we can resample and export from ILWIS any result of spatial prediction, e.g. by using:

```
TEXT_kml.mpr{dom=Color.dom} = MapResample(TEXT,geo1s.grf,bilinear)
export BMP(text_kml.mpr,text_kml)
```

where geo1s.grf is the 1 arsec grid definition with the parameters listed above and MapResample is the ILWIS function to resample a raster map using bilinear method. We can copy the exported image to a server and put its URL into the KML file. The final layout can be seen in Fig. 4.28. This is now a complete visualization that shows both the locations of sampled values, final results of spatial prediction and the associated legend.

In R, any point or line/polygon sp dataset can be exported to KML by using the writeOGR method of the rgdal package. In the case of Ebergötzen, you can import the points as a table, then make a sp layer and then do all the calculations with it even if there is no geographic projection attached to it. However, if you want to reproject the
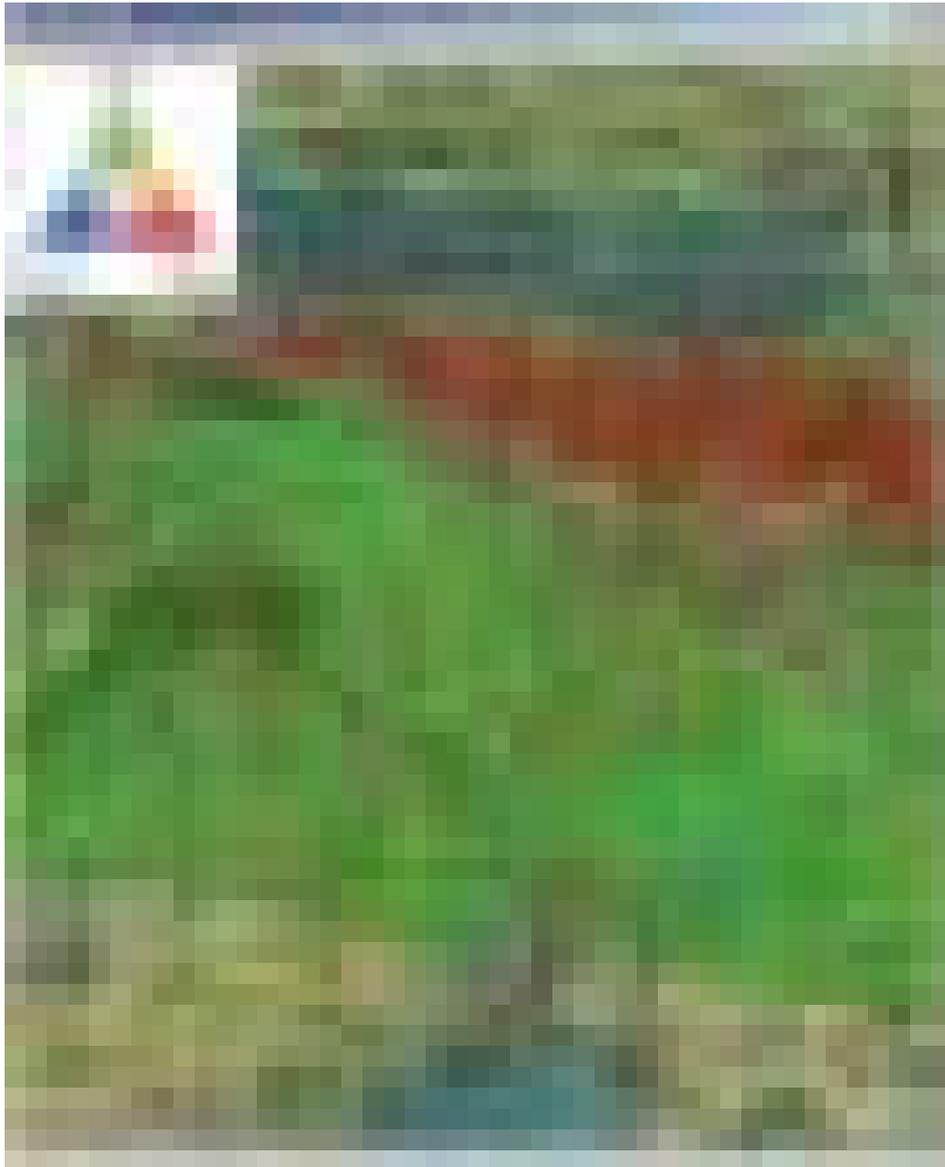
Fig. 4.28: The final Google Earth layout showing predicted soil texture fractions in topsoil.

sp layers to another coordinate system, you need to consider giving the data the correct projection string as demonstrated in §4.2.1. Once the correct coordinate system has been attached, we can reproject any point or grid layer to the longlat system:

```
SAND.rklonglat = spTransform(SAND.rk["pred"], CRS("+proj=longlat"))
```

which might take some time to calculate. Note that sp package will convert the raster map to a point map, which means that we need to convert this map again to a sp grid dataframe before we can export it. First, we need to define the new geographic grid system with the cell size of 0.000278:

```
geo1sec = spsample(SAND.rklonglat, type="regular",
             cellsize=c(0.000278,0.000278))
gridded(geo1sec) = TRUE
```

We can see that sp created the following new grid definition:

```
> gridparameters(geo1sec)
   cellcentre.offset cellsize cells.dim
x1          10.00751 0.000278       524
x2          51.50106 0.000278       327
```

This is an empty grid without any topology (only grid nodes are defined) and co-ordinate system definition. To create topology, we coerce a dummy variable (1s), then specify that the layer has a full topology:

```
geo1sec$v = rep(1, geo1sec@grid@cells.dim["x1"]*
                   geo1sec@grid@cells.dim["x2"])
fullgrid(geo1sec)=TRUE
proj4string(geo1sec) = CRS("+proj=longlat")
```

and estimate the values of the reprojected map at new grid locations using the bilinear resampling:

```
SAND.rklonglatg = krige(pred~1, SAND.rklonglat, geo1sec, nmax=4)
spplot(geo1sec["SAND"])
```

The final grid map can be exported to KML format using the maptools package and kmlOverlay method:

```
SAND.rkkml = GE_SpatialGrid(SAND.rklonglatg)
writeGDAL(SAND.rklonglatg[1], "SANDrk.tif",
          drivername="GTiff", type="Byte")
kmlOverlay(SAND.rkkml, kmlfile="SANDrk.kml",
          imagefile="SANDrk.tif", name="SAND in %")
```

which will automatically generate a KML file with an ground overlay. In this case we do not have much options to change the legend of the geotiffs. The whole process of resampling the grids in R can be quite time consuming, so I can only advise you to instead run the resampling in ILWIS and use R only to run statistical analysis and make predictions/simulations.

Alternatively, you can also export a PNG of an R plot (Fig. 4.29). First create a temporary file and paste an empty PNG image which is the same size as the GE SpatialGrid:

```
tf <- tempfile()
png(file=paste(tf, ".png", sep=""), width=SAND.rkkml$width,
    height=SAND.rkkml$height, bg="transparent")
```

now plot the output map as an image and (optional) overlay the location of the sampling locations (points):

```
par(mar=c(0,0,0,0), xaxs="i", yaxs="i")
image(as.image.SpatialGridDataFrame(SAND.rklonglatg[1]), col=bpy.colors(),
  xlim=SAND.rkkml$xlim, ylim=SAND.rkkml$ylim)
plot(points.longlat, pch="+", cex=1.2, add=TRUE, bg="transparent")
```

note that we need to define the margins (mar=c(0,0,0,0)) of the plotted image to equal zero, this way the coordinates of the plotted image correspond to the geographic coordinates of the map. Finally, we export the KML overlay using:
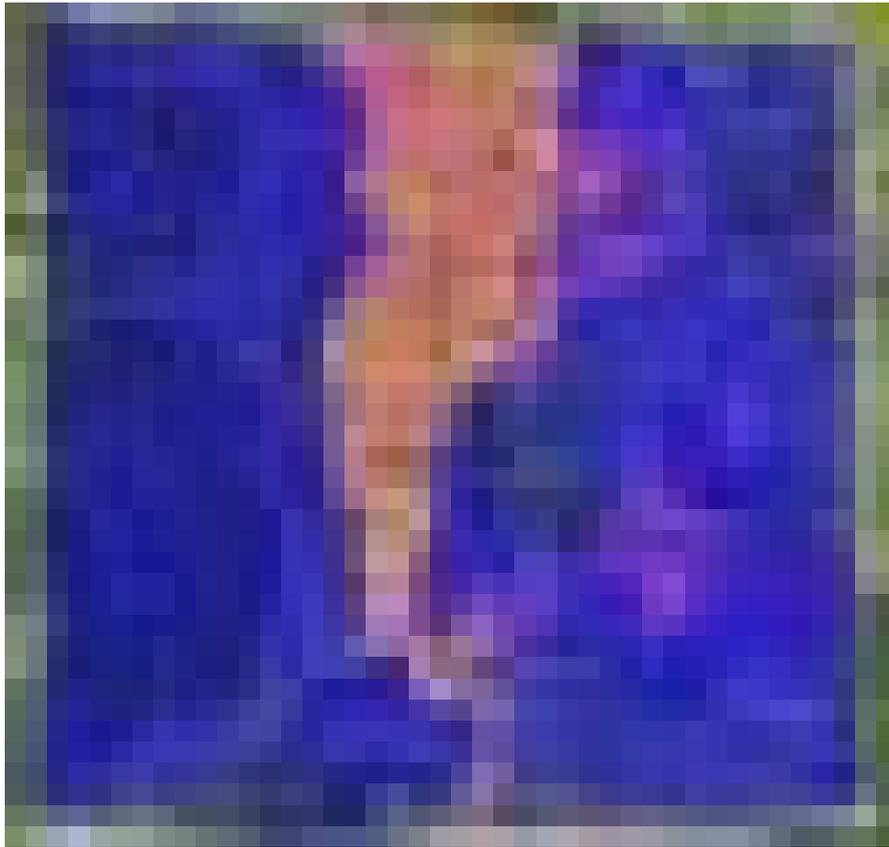
Fig. 4.29: R plot exported to Google Earth. Compare with Fig. 4.23.

```
kmlOverlay(SGqk, paste(tf, ".kml", sep=""), paste(tf, ".png", sep=""))
dev.off()
```

The final KML file/PNG image might need some further manual editing, e.g. to place the PNG file on some server or add more description about how was the map produced and where to find the original data.

4.8.3   Alternative ways to geovisualization

The users of MatLab can explore possibilities of exporting the results of geostatistical analysis to Google Earth by using the Google Earth Toolbox. This toolbox allows not only export of raster maps (ground overlays) but also a friendly tool to export the associated legends, generate 3D surfaces, contours from isometric maps, wind barbs and 3D vector objects. Once a map has produced using some spatial prediction technique, it can converted to a KML format using e.g.:

```
examplemap = ge_groundoverlay(N,E,S,W,... 'imageURL','map.bmp');

ge_output('examplemap.kml',kmlStr);
```

where N, E, S, W are the bounding coordinates that can be determined automatically or set by the user.

Another sophisticated option to visualize the results of (spatio-temporal) geostatistical mapping is to use a small stand-alone visualization software called Aquila (Pebesma

et al., 2007). Aquila allows interactive exploration of the spatio-temporal **Cumulative Distribution Functions** (CDFs) and allows decision makers to explore uncertainty associated to attaching different threshold or its spatial distribution in the area of interest. It is actually rather simple to use — one only needs to prepare a sample (e.g. 12 slices) of quantile estimates, which are then locally interpolated to produce CDFs.

Important sources:

★ Minasny, B. and McBratney, A. B., 2007. Spatial prediction of soil properties using EBLUP with Matérn covariance function. Geoderma, 140: 324–336.

★ Hengl T., Toomanian N., Reuter H. I., Malakouti M. J. 2007. Methods to interpolate soil categorical variables from profile observations: lessons from Iran. Geoderma, 140(4): 417–427.

★ Böhner, J., McCloy, K. R. and Strobl, J. (Eds), 2006. SAGA — Analysis and Modelling Applications. Göttinger Geographische Abhandlungen, Heft 115. Verlag Erich Goltze GmbH, Göttingen, 117 pp.

★ Pebesma, E. J., 2006. The Role of External Variables and GIS Databases in Geostatistical Analysis. Transactions in GIS, 10(4): 615–632.

★ Hengl T., Toomanian N., 2006. Maps are not what they seem: representing uncertainty in soil-property maps. In: Caetano, M., Painho, M., (eds) Proceedings of the 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences (Accuracy 2006), 5-7 July 2006, Lisbon, Portugal, pp. 805–813.

★ http://math.uni-klu.ac.at/stat/ — Department of Statistics in Klagenfurt regularly organizes international workshop/conference "*Interfacing Geostatistics, GIS and Spatial Databases*" (statGIS).

Bibliography

Ahmed, S., de Marsily, G., 1987. Comparison of geostatistical methods for estimating transmissivity using data on transmissivity and specific capacity. Water Resources Research 23 (9): 1717–1737.

Antonić, O., Kušan, V., Bakran-Petricioli, T., Alegro, A., Gottstein-Matočec, S., Peternel, H., Tkalčec, Z., 2005. Mapping the habitats of The Republic of Croatia (2000.-2004.) — The project overview (in Croatian). Drypis — Journal for Applied Ecology 1 (1): 40.

Atkinson, P., Quattrochi, D. A., 2000. Special issue on geostatistics and geospatial techniques in remote sensing. Computers & Geosciences 26 (4): 359.

Baddeley, A., Turner, R., 2005. Spatstat: an R package for analyzing spatial point patterns. Journal of Statistical Software 12 (6): 1–42.

Bailey, N., Clements, T., Lee, J. T., Thompson, S., 2003. Modelling soil series data to facilitate targeted habitat restoration: a polytomous logistic regression approach. Journal of Environmental Management 67 (4): 395–407.

Banks, J. (Ed.), 1998. Handbook of Simulation — Principles, Methodology, Advances, Applications, and Practice. Wiley, New York, p. 864.

Bierkens, M. F. P., Burrough, P. A., 1993. The indicator approach to categorical soil data I: Theory. Journal of Soil Science 44: 361–368.

Bishop, T. F. A., McBratney, A. B., 2001. A comparison of prediction methods for the creation of field-extent soil property maps. Geoderma 103 (1-2): 149–160.

Bishop, T. F. A., Minasny, B., 2005. Digital Soil-Terrain Modelling: The Predictive Potential and Uncertainty. In: Grunwald, S. (Ed.), Environmental Soil-Landscape Modeling: Geographic Information Technologies and Pedometrics. CRC Press, Boca Raton, Florida, pp. 185–213.

Bivand, R. S., 2005. Interfacing GRASS 6 and R. Status and development directions. GRASS Newsletter 3: 11–16.

Bleines, C., Perseval, S., Rambert, F., Renard, D., Touffait, Y., 2004. ISATIS. Isatis software manual, 5th Edition. Geovariances & Ecole Des Mines De, Paris, p. 710.

Bonan, G. B., Levis, S., Sitch, S., Vertenstein, M., Oleson, K. W., 2003. A dynamic global vegetation model for use with climate models: concepts and description of simulated vegetation dynamics. Global Change Biology 9 (11): 1543–1566.

Boots, B. N., Getis, A., 1988. Point pattern analysis. Scientific Geography series 8. Sage Publications, Newbury Park, p. 93.

Boucneau, G., van Meirvenne, M., Thas, O., Hofman, G., 1998. Integrating properties of soil map delineations into ordinary kriging. European Journal of Soil Science 49 (2): 213–229.

Bragato, G., 2004. Fuzzy continuous classification and spatial interpolation in conventional soil survey for soil mapping of the lower Piave plain. Geoderma 118 (1-2): 1–16.

Brus, D. J., Heuvelink, G. B. M., 2007. Optimization of sample patterns for universal kriging of environmental variables. Geoderma 138 (1-2): 86–95.

Burrough, P. A., McDonnell, R. A., 1998. Principles of Geographical Information Systems. Oxford University Press Inc., New York, p. 333.

Carré, F., Girard, M. C., 2002. Quantitative mapping of soil types based on regression kriging of taxonomic distances with landform and land cover attributes. Geoderma 110 (3-4): 241–263.

Chambers, J. M., Hastie, T. J., 1992. Statistical Models in S. Wadsworth & Brooks/Cole, Pacific Grove, California, p. 595.

Chiles, J. P., Delfiner, P., 1999. Geostatistics: modeling spatial uncertainty. John Wiley & Sons, New York.

Christensen, R., 2001. Linear Models for Multivariate, Time Series, and Spatial Data, 2nd Edition. Springer Verlag, New York, p. 393.

Congalton, R. G., Green, K., 1999. Assessing the accuracy of remotely sensed data: principles and practices. Lewis, Boca Raton, FL, p. 137.

Conrad, O., 2006. SAGA — Program Structure and Current State of Implementation. In: Böhner, J., McCloy, K. R., Strobl, J. (Eds.), SAGA — Analysis and Modelling Applications. Vol. 115. Verlag Erich Goltze GmbH, pp. 39–52.

Cressie, N., 1990. The origins of kriging. Mathematical Geology 22 (3): 239–252.

Cressie, N. A. C., 1993. Statistics for Spatial Data, revised edition. John Wiley & Sons, New York, p. 416.

D'Agostino, V., Zelenka, A., 1992. Supplementing solar radiation network data by co-Kriging with satellite images. International Journal of Climatology 12 (7): 749–761.

de Gruijter, J. J., Walvoort, D. J. J., van Gaans, P. F. M., 1997. Continuous soil maps — a fuzzy set approach to bridge the gap between aggregation levels of process and distribution models. Geoderma 77 (2-4): 169–195.

Deutsch, C. V., Journel, A. G., 1998. GSLIB: Geostatistical Software and User's Guide, 2nd Edition. Oxford University Press, New York.

Dooley, M. A., Lavin, S. J., 2007. Visualizing method-produced uncertainty in isometric mapping. Cartographic Perspectives 56: 17–36.

D'Or, D., 2003. Spatial prediction of soil properties, the Bayesian Maximum Entropy approach. Phd, Université Catholique de Louvain.

D'Or, D., Bogaert, P., 2005. Spatial prediction of categorical variables with the Bayesian Maximum Entropy approach: the Ooypolder case study. European Journal of Soil Science 55 (December): 763–775.

Draper, N. R., Smith, H., 1998. Applied Regression Analysis, 3rd Edition. John Wiley, New York, p. 697.

Dubois, G. (Ed.), 2005. Automatic mapping algorithms for routine and emergency monitoring data. Report on the Spatial Interpolation Comparison (SIC2004) exercise. EUR 21595 EN. Office for Official Publications of the European Communities, Luxembourg, p. 150.

Dubois, G., Galmarini, S., 2004. Introduction to the Spatial Interpolation Comparison (SIC). Applied GIS 1 (2): 9–11.

Foody, G. M., 2004. Thematic map comparison: evaluating the statistical significance of differences. Photogrammetric Engineering and Remote Sensing 70: 627–633.

Fotheringham, A. S., Brunsdon, C., Charlton, M., 2002. Geographically Weighted Regression: The Analysis of Spatially Varying Relationships. GIS & Remote Sensing. Wiley, p. 282.

Gandin, L. S., 1963. Objective Analysis of Meteorological Fields. translated from Russian in 1965 by Israel Program for Scientific Translations, Jerusalem. Gidrometeorologicheskoe Izdatel'stvo (GIMIZ), Leningrad, p. 242.

Gehrt, E., Böhner, J., 2001. Vom punkt zur flache — probleme des "upscaling" in der bodenkartierung. In: Diskussionsforum Bodenwissenschaften: Vom Bohrstock zum Bildschirm. FH, Osnabrück, pp. 17–34.

Goovaerts, P., 1997. Geostatistics for Natural Resources Evaluation. Oxford University Press, New York, p. 496.

Goovaerts, P., 1999. Geostatistics in soil science: State-of-the-art and perspectives. Geoderma 89 (1-2): 1–45.

Gotway, C. A., Stroup, W. W., 1997. A Generalized Linear Model approach to spatial data analysis and prediction. Journal of Agricultural, Biological, and Environmental Statistics 2 (2): 157–198.

Grose, D., Crouchley, R., van Ark, T., Allan, R., Kewley, J., Braimah, A., Hayes, M., 2006. sabreR: Grid-Enabling the Analysis of MultiProcess Random Effect Response Data in R. In: Halfpenny, P. (Ed.), Second International Conference on e-Social Science. Vol. 3c. National Centre for e-Social Science, Manchester, UK, p. 12.

Haas, T. C., 1990. Kriging and automated semivariogram modelling within a moving window. Atmospheric Environment 24A: 1759–1769.

Hardy, R. L., 1971. Multiquadratic equations of topography and other irregular surfaces. Journal of Geophysical Research 76: 1905–1915.

Henderson, B. L., Bui, E. N., Moran, C. J., Simon, D. A. P., 2004. Australia-wide predictions of soil properties using decision trees. Geoderma 124 (3-4): 383–398.

Henderson, B. L., Bui, E. N., Moran, C. J., Simon, D. A. P., 2005. Australia-wide predictions of soil properties using decision trees. Geoderma 124 (3-4): 383–398.

Hengl, T., 2006. Finding the right pixel size. Computers & Geosciences 32 (9): 1283–1298.

Hengl, T., Bajat, B., Reuter, H. I., Blagojević, 2007a. Geostatistical modelling of topography using auxiliary maps. Computers and Geosciences in review.

Hengl, T., Heuvelink, G. B. M., Rossiter, D. G., 2007b. About regression-kriging: from theory to interpretation of results. Computers & Geosciences in press.

Hengl, T., Heuvelink, G. M. B., Stein, A., 2004a. A generic framework for spatial prediction of soil variables based on regression-kriging. Geoderma 122 (1-2): 75–93.

Hengl, T., Rossiter, D. G., Stein, A., 2004b. Soil sampling strategies for spatial prediction by correlation with auxiliary maps. Australian Journal of Soil Research 41 (8): 1403–1422.

Hengl, T., Toomanian, N., 2006. Maps are not what they seem: representing uncertainty in soil-property maps. In: Caetano, M., Painho, M. (Eds.), Proceedings of the 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences (Accuracy 2006). Instituto Geográphico Português, Lisbon, Portugal, pp. 805–813.

Hengl, T., Toomanian, N., Reuter, H. I., Malakouti, M. J., 2007c. Methods to interpolate soil categorical variables from profile observations: lessons from Iran. Geoderma 140 (4): 417–427.

Hession, S. L., Shortridge, A. M., Torbick, M. N., 2006. Categorical models for spatial data uncertainty. In: Caetano, M., Painho, M. (Eds.), Proceedings of the 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences (Accuracy 2006). Instituto Geográphico Português, Lisbon, pp. 386–395.

Heuvelink, G., 1998. Error propagation in environmental modelling with GIS. Taylor & Francis, London, UK, p. 144.

Heuvelink, G. B. M., Pebesma, E. J., 1999. Spatial aggregation and soil process modelling. Geoderma 89 (1-2): 47–65.

Heuvelink, G. B. M., Webster, R., 2001. Modelling soil variation: past, present, and future. Geoderma 100 (3-4): 269–301.

Hiemstra, P. H., Pebesma, E. J., Twenhöfel, C. J. W., Heuvelink, G. B. M., 2007. Toward an automatic real-time mapping system for radiation hazards. In: Klien, E. (Ed.), GI-Days conference. Institut für Geoinformatik, Münster, Germany, p. 6.

Hutchinson, M. F., 1995. Interpolating mean rainfall using thin plate smoothing splines. International Journal of Geographical Information Systems 9: 385–403.

Isaaks, E. H., Srivastava, R. M., 1989. Applied Geostatistics. Oxford University Press, New York, p. 542.

Jarvis, C. H., Stuart, N., 2001. A comparison among strategies for interpolating maximum and minimum daily air temperatures. Part II: The interaction between number of guiding variables and the type of interpolation method. Journal of Applied Meteorology 40: 1075–1084.

Journel, A. G., 1986. Constrained interpolation and qualitative information. Mathematical Geology 18 (3): 269–286.

Kanevski, M., Maignan, M., Demyanov, V., Maignan, M., 1997. How neural network 2-d interpolations can improve spatial data analysis: neural network residual kriging (nnrk). In: Hohn, M. (Ed.), Proceedings of the Third Annual Conference of the IAMG. International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, pp. 549–554.

Kitanidis, P. K., 1994. Generalized covariance functions in estimation. Mathematical Geology 25: 525–540.

Knotters, M., Brus, D. J., Voshaar, J. H. O., 1995. A comparison of kriging, co-kriging and kriging combined with regression for spatial interpolation of horizon depth with censored observations. Geoderma 67 (3-4): 227–246.

Krige, D. G., 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. Journal of the Chemical, Metallurgical and Mining Society 52: 119–139.

Kuo, W. L., Steenhuis, T. S., McCulloch, C. E., Mohler, C. L., Weinstein, D. A., DeGloria, S. D., Swaney, D. P., 1999. Effect of grid size on runoff and soil moisture for a variable-source-area hydrology model. Water Resources Research 35 (11): 3419–3428.

Kyriakidis, P. C., Journel, A. G., 1999. Geostatistical Space–Time Models: A Review. Mathematical Geology 31 (6): 651–684.

Kyriakidis, P. C., Shortridge, A. M., Goodchild, M. F., 1999. Geostatistics for conflation and accuracy assessment of Digital Elevation Models. International Journal of Geographical Information Science 13 (7): 677–708.

Lam, N. S.-N., 1983. Spatial interpolation methods: a review. The American Cartographer 10: 129–149.

Lark, R. M., Cullis, B., Welham, S. J., 2005. On Spatial Prediction of Soil Properties in the Presence of a Spatial Trend: The Empirical Best Linear Unbiased Predictor (E-BLUP) with REML. European Journal of Soil Science 57: 787–799.

Latimer, A. M., Wu, S., Gelfand, A. E., Silander Jr., J. A., 2004. Building statistical models to analyze species distributions. Ecological Applications 16 (1): 33–50.

Leopold, U., Heuvelink, G. B. M., Tiktak, A., Finke, P. A., Schoumans, O., 2005. Accounting for change of support in spatial accuracy assessment of modelled soil mineral phosphorous concentration. Geoderma 130 (3-4): 368–386.

Li, W., Zhang, C., Burt, J., Zhu, A., 2005. A markov chain-based probability vector approach for modelling spatial uncertainties of soil classes. Soil Science Society of America Journal 69: 1931–1942.

Li, W., Zhang, C., Burt, J., Zhu, A., Feyen, J., 2004. Two-dimensional markov chain simulation of soil type spatial distribution. Soil Science Society of America Journal 68: 1479–1490.

Lloyd, C. D., 2005. Assessing the effect of integrating elevation data into the estimation of monthly precipitation in Great Britain. Journal of Hydrology 308 (1-4): 128–150.

Matheron, G., 1962. Traité de géostatistique appliquée. Vol. 14 of Mémoires du Bureau de Recherches Géologiques et Minières. Editions Technip, Paris, p. NA.

Matheron, G., 1969. Le krigeage universel. Vol. 1. Cahiers du Centre de Morphologie Mathématique, École des Mines de Paris, Fontainebleau, p. NA.

McBratney, A. B., 1998. Some considerations on methods for spatially aggregating and disaggregating soil information. Nutrient Cycling in Agroecosystems 50: 51–62.

McBratney, A. B., de Gruijter, J. J., Brus, D. J., 1992. Spatial prediction and mapping of continuous soil classes. Geoderma 54 (1-4): 39–64.

McBratney, A. B., Mendoça Santos, M. L., Minasny, B., 2003. On digital soil mapping. Geoderma 117 (1-2): 3–52.

McKenzie, N. J., Ryan, P. J., 1999. Spatial prediction of soil properties using environmental correlation. Geoderma 89 (1-2): 67–94.

Minasny, B., McBratney, A. B., 2001. A rudimentary mechanistic model for soil formation and landscape development II. A two-dimensional model incorporating chemical weathering. Geoderma 103: 161–179.

Minasny, B., McBratney, A. B., 2005. The Matérn function as a general model for soil variograms. Geoderma 128 (3-4): 192–207.

Minasny, B., McBratney, A. B., 2006. A conditioned Latin hypercube method for sampling in the presence of ancillary information. Computers & Geosciences 32 (9): 1378–1388.

Minasny, B., McBratney, A. B., 2007. Spatial prediction of soil properties using EBLUP with Matérn covariance function. Geoderma 140: 324–336.

Mitas, L., Mitasova, H., 1999. Spatial interpolation. In: Longley, P., Goodchild, M. F., Maguire, D. J., Rhind, D. W. (Eds.), Geographical Information Systems: Principles, Techniques, Management and Applications. Vol. 1. Wiley, pp. 481–492.

Mitášová, H., Mitas, L., 1993. Interpolation by regularized spline with tension, I Theory and implementation. Mathematical Geology 25: 641–655.

Mitasova, H., Mitas, L., Russell, S. H., 2005. Simultaneous spline interpolation and topographic analysis for lidar elevation data: methods for Open source GIS. IEEE Geoscience and Remote Sensing Letters 2 (4): 375–379.

Murrell, P., 2006. R Graphics. Computer Science and Data Analysis Series. Chapman & Hall/CRC, Boca Raton, FL, p. 328.

Myers, D. E., 1994. Spatial interpolation: an overview. Geoderma 62: 17–28.

Neter, J., Kutner, M. H., Nachtsheim, C. J., Wasserman, W. (Eds.), 1996. Applied Linear Statistical Models, 4th Edition. McGraw-Hill, p. 1391.

Nielsen, D., Wendroth, O., 2003. Spatial and Temporal Statistics — Sampling Field Soils and Their Vegetation. GeoEcology textbook. Catena-Verlag, Reiskirchen, p. 614.

Odeh, I. O. A., McBratney, A. B., Chittleborough, D. J., 1995. Further results on prediction of soil properties from terrain attributes: heterotopic cokriging and regression-kriging. Geoderma 67 (3-4): 215–226.

Olaya, V. F., 2004. A gentle introduction to SAGA GIS. The SAGA User Group e.V., Gottingen, Germany, p. 208.

Ott, R. L., Longnecker, M. (Eds.), 2001. An Introduction to Statistical Methods and Data Analysis, 5th Edition. Duxbury press, p. 1152.

Papritz, A., Stein, A., 1999. Spatial prediction by linear kriging. In: Stein, A., van der Meer, F., Gorte, B. (Eds.), Spatial statistics for remote sensing. Kluwer Academic publishers, Dodrecht, pp. 83–113.

Pardo-Iguzquiza, E., Dowd, P. A., 2005. Multiple indicator cokriging with application to optimal sampling for environmental monitoring. Computers & Geosciences 31 (1): 1–13.

Park, S. J., Vlek, P. L. G., 2002. Environmental correlation of three-dimensional soil spatial variability: a comparison of three adaptive techniques. Geoderma 109 (1-2): 117–140.

Patil, G. P., 2002. Composite sampling. In: El-Shaarawi, A. H., Piegorsch, W. W. (Eds.), Encyclopedia of Environmetrics. Vol. 1. John Wiley & Sons, Chichester, UK, pp. 387–391.

Pebesma, E. J., 2004. Multivariable geostatistics in S: the gstat package. Computers & Geosciences 30 (7): 683–691.

Pebesma, E. J., Bivand, R. S., 2005. Classes and methods for spatial data in R. R News 5 (2): 913.

Pebesma, E. J., de Jong, K., Briggs, D. J., 2007. Visualising uncertain spatial and spatio-temporal data under different scenarios: an air quality example. International Journal of Geographical Information Science 21 (5): 515527.

Pebesma, E. J., Duin, R. N. M., Burrough, P. A., 2005. Mapping sea bird densities over the North Sea: spatially aggregated estimates and temporal changes. Environmetrics 16 (6): 573–587.

Rossiter, D. G., 2007a. Introduction to the R Project for Statistical Computing for use at ITC, 2nd Edition. International Institute for Geo-information Science & Earth Observation (ITC), Enschede, Netherlands, p. 136.

Rossiter, D. G., 2007b. Technical Note: Co-kriging with the gstat package of the R environment for statistical computing, 2nd Edition. International Institute for Geo-information Science & Earth Observation (ITC), Enschede, Netherlands, p. 81.

Rowe, J. S., Barnes, B. V., 1994. Geo-ecosystems and bio-ecosystems. Bulletin of the Ecological Society of America 75 (1): 40–41.

Rykiel, E. J., 1996. Testing ecological models: the meaning of validation. Ecological Modelling 90: 229–244.

Schoorl, J. M., Veldkamp, A., Bouma, J., 2002. Modelling water and soil redistribution in a dynamic landscape context. Soil Science Society of America Journal 66 (5): 1610–1619.

Shamoun, J. Z., Sierdsema, H., van Loon, E. E., van Gasteren, H., Bouten, W., Sluiter, F., 2005. Linking Horizontal and Vertical Models to Predict 3D + time Distributions of Bird Densities. International Bird Strike Committee, Athens, p. 11.

Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. In: Blue, R. B. S., Rosenberg, A. M. (Eds.), Proceedings of the 1968 ACM National Conference. ACM Press, New York, p. 517524.

Skaggs, T. H., Arya, L. M., Shouse, P. J., Mohanty, B. P., 2001. Estimating Particle-Size Distribution from Limited Soil Texture Data. Soil Science Society of America Journal 65 (4): 1038–1044.

Stein, A., Riley, J., Halberg, N., 2001. Issues of scale for environmental indicators. Agriculture, Ecosystems & Environment 87 (2): 215–232.

Stein, M. L., 1999. Interpolation of Spatial Data: Some Theory for Kriging. Series in Statistics. Springer, New York, p. 247.

Szalai, S., Bihari, Z., Szentimrey, T., Lakatos, M. (Eds.), 2007. COST Action 719 — The Use of Geographic Information Systems in Climatology and Meteorology. Proceedings from the Conference on on spatial interpolation in climatology and meteorology. Office for Official Publications of the European Communities, Luxemburg, p. 264.

Thompson, J. A., Bell, J. C., Butler, C. A., 2001. Digital elevation model resolution: effects on terrain attribute calculation and quantitative soil-landscape modeling. Geoderma 100: 67–89.

Triantafilis, J., Ward, W. T., Odeh, I. O. A., McBratney, A. B., 2001. Creation and Interpolation of Continuous Soil Layer Classes in the Lower Namoi Valley. Soil Science Society of America Journal 65: 403–413.

Unit Geo Software Development, 2001. ILWIS 3.0 Academic user's guide. ITC, Enschede, p. 520.

Venables, W. N., Ripley, B. D., 2002. Modern applied statistics with S, 4th Edition. Springer-Verlag, New York, p. 481.

Wackernagel, H., 2003. Multivariate geostatistics: an introduction with applications, 2nd Edition. Springer-Verlag, p. 381.

Walter, C., McBratney, A. B., Donuaoui, A., Minasny, B., 2001. Spatial prediction of topsoil salinity in the Chelif valley, Algeria, using local ordinary kriging with local variograms versus whole-area variogram. Australian Journal of Soil Research 39: 259–272.

Walvoort, D. J. J., de Gruijter, J. J., 2001. Compositional Kriging: A Spatial Interpolation Method for Compositional Data. Mathematical Geology 33 (8): 951–966.

Webster, R., Oliver, M. A., 2001. Geostatistics for Environmental Scientists. Statistics in Practice. Wiley, Chichester, p. 265.

Wood, J., 2007. Overview of software packages used in geomorphometry. In: Hengl, T., Reuter, H. I. (Eds.), Geomorphometry: concepts, software, applications. Developments in Soil Science. Elsevier, p. in press.

Xiao, J., Shen, Y., Tateishi, R., Bayaer, W., 2006. Development of topsoil grain size index for monitoring desertification in arid land using remote sensing. International Journal of Remote Sensing 27 (12): 2411–2422.

Zhou, F., Huai-Cheng, G., Yun-Shan, H., Chao-Zhong, W., 2007. Scientometric analysis of geostatistics using multivariate methods. Scientometrics in press.

Index