Rozšíření interpolačních nástrojů v R Project o modely nejistoty

Tomáš Burian

Katedra geoinformatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci,
17. listopadu 1192/12, 771 46, Olomouc, ČR

Buri777@seznam.cz

Abstrakt.

Práce se zabývá praktickým rozšiřováním existujících interpolačních funkcí v softwaru R project o nejistotu vstupních dat. Zmíněné rozšíření představuje možnost vytvořit několik rozdílných modelů nejistoty nad vstupními daty. Touto implementací se rozšiřují možnosti ve vyjádření vstupních dat, a prohlubují se možnosti interpolačních procesů a také i komplexnost výstupních dat. Samotnou nejistotu dat lze vytvořit na základě očekávané neurčitosti zkoumané proměnné. Tato neurčitost může být vyjádřena ve formě prostorově korelované chyby, náhodnými hodnotami, procentem hodnoty, konstantou či náhodnou odchylkou od původních dat. Vytvořené modely nejistoty jsou postaveny na possibilistickém vyjádření dat, nikoli na statistice. Upravené záznamy pak vstupují do interpolací a vzniká model proměnné obsahující i dodanou nejistotu. Po ukončení interpolace je možné získané výsledky dále upravovat a zpracovávat v rámci softwaru R, čímž lze dále vhodně rozšiřovat výsledky této bakalářské práce.

Výsledkem celé práce pak je balíček pro R, jehož obsahem jsou funkce pro kompletní tvorbu příkladných dat včetně jejich úpravy do vytyčeného formátu. Dále celkem 6 funkcí pro tvorbu modelů nejistoty. Tři druhy základních interpolací (spline, kriging, IDW) a funkce k vytváření gridů pro vstup nejen do interpolací.

Hlavním cílem této bakalářské práce bylo otestovat možnosti R v interpolacích obohacených o nejistotu. Jedná se tedy o první práci svého druhu, kterou je možné dále vyvíjet a rozšiřovat. Z autorského hlediska se jednalo především o vytvoření pomyslného prvního kroku, zjištění faktických informací a vytvoření základního kamene v této doposud nedostatečně probádané oblasti.

Klíčová slova: R project, nejistota, model, interpolace.

Abstract. The work deals with the practical expansion of existing interpolation functions in software R project by using the uncertainty of input data. These extensions are represented in the ability to create several different models of uncertainty over the input data. The implementation will expand the possibilities in declarations of input data, improve the possibilities of interpolation processes, and also the complexity of the output data. The actual data uncertainty can be based on the expected vagueness of the variable. This vagueness can be putted into the form of spatially correlated errors, random values, percentage of values, constant or random deviations from the original data. Created uncertainty models are built on possibilistic

expression of data, not on statistics. After that the data are going into the interpolations processes that creates model of the variable containing the uncertainty of the input. After the end of interpolation it is possible to keep on working and process the obtained results in the software R, what could be the possible way how to improve the results of this bachelor thesis.
The result of the entire work is a package for R, that contents functions to provide the exemplary data creation, including adjustments to the stated format. Additionally 6 functions for creating the models of uncertainty were programmed. Three types of basic interpolations (spline, kriging, IDW) and a function to create grids for the interpolations.
The main objective of this bachelor thesis was to test the options of R interpolation methods enriched by the uncertainty. This is therefore the first work of its kind that can be further developed and expanded. From the author's point of view, it was all about creating an imaginary first step, finding the actual informations and creating of the foundation stone in this so far under-explored area.

Keywords: R project, uncertainty, model, interpolation.

1 Úvod

Problematika nejistoty je velmi často přehlížena. Lidé mnohdy ani netuší, že se s tímto fenoménem setkávají. Dokonce lze konstatovat, že nás obklopuje neustále, ať už v profesním či soukromém životě. Člověk má jistý odhad pro řadu popsatelných jevů, tento odhad navíc dokáže ještě upřesnit různými postupy či technologiemi. Avšak nakolik si můžeme být jisti, že je aplikovaná technika naprosto přesná? Existují ověřené teorie, které dokazují právě nepřesnost a tím i potvrzují existenci nejistoty výsledných dat. Nepřesné mohou být geodetická měření, kde se téměř vždy vyskytuje určitá odchylka od reality. Také metody sběru výškových dat, při snímání povrchu Země, vykazují zřejmou nejistotu v rámci až několika metrů.

Je tedy očividné, že nejistota opravdu existuje. Prof. Mikhail Kanevski v roce 2012 prohlásil, že je tato oblast nedostatečně prozkoumána a právě proto by bylo velice příhodné využít vyspělosti dnešního světa a aplikovat nejistotu všude tam, kde to jen lze. Samotné modely nejistoty jsou poté jednou z možností, jak toto tvrzení podpořit například v oblasti modelování zemského povrchu.

Spojením modelů nejistot a interpolačních nástrojů lze prakticky ukázat, jaké rozdíly ve výsledku způsobují variabilní

vstupní data. Proto je nutné uvědomit si, že ne vždy je získaná hodnota ta správná a že může být ovlivněna nejistotou.

2 Postup zpracování

Na počátku celé práce bylo nutné nejprve nastudovat patřičnou literaturu, a to především v oblasti problematiky nejistoty. Dalším důležitým bodem při studování teorie k bakalářské práci byla i oblast programování v R a studium žádaných balíčků pro R project. Na základě získaných znalostí byly vytvářeny rešerše vysvětlující základní poznatky a teorie v rámci jednotlivých sekcí.

Po získání potřebných informací a teoretického základu, bylo přistoupeno ke tvorbě praktické části. Úspěšné a vyhovující podobě algoritmů a klíčových funkcí předcházelo samozřejmě sekundární studium zdrojových kódů a principů již vytvořených funkcí. Veškeré nabyté dojmy, proč použít právě R, pak potvrzuje myšlenka (Matloff, 2011): je krásné a levné, proč používat něco jiného? Přistoupilo se tedy k fázi programování nejprve funkcí pro generování dat a gridu. Vstupní data dostala základní atributy *x, y, z* a požadovaný typ třídy objektu *uncertainSpatialPoints*. Grid byl vytvořen na základě zvolených parametrů požadovaného rozměru. Pro úplnost procesu byl na počátku naprogramován i první model nejistoty, který se aplikoval na primární data a modifikoval je tak pro vstup do interpolačních funkcí. Posléze byly takticky naprogramovány interpolační funkce, na kterých se testovala funkčnost právě vygenerovaných dat a zároveň i správnost interpolačních procesů. Celkem byly zvoleny 3 druhy interpolací, a to metoda IDW, spline a kriging. Jako nástavba byla sestavena i funkce pro odhad variogramu, jenž vstupuje do kriging interpolace. Tato dodatečná procedura je určena spíše pro náročnější uživatele, kteří mají zájem o hlubší pohled na algoritmizaci krigingu. Do každé funkce vstupovaly vygenerovaná data a po dokončení průběhu byl výsledkem nový objekt, se kterým lze dále pracovat.

Po vyřešení problematiky vstupu dat a interpolací byly vytvořeny modely nejistot, které lze aplikovat na jakákoli vstupní

data požadovaného formátu. Určení nejistoty probíhá podle definovaných možností, jako například procentuální přesnost dat, náhodné hodnoty v intervalu či s využitím náhodné odchylky. V závěru praktické části byly všechny tyto funkce zabaleny do výsledného balíčku, který je možné volně připojit do softwaru RStudio. Balíček bude mít celkem 6 složek, v nichž se bude ukrývat několik užitečných funkcí pro tvorbu:

- dat,
- modelů nejistoty,
- požadované třídy objektů,
- gridu,
- interpolace (IDW, Spline, Kriging)
- variogramu.

3 Balíček UNCERTAINTYINTERPOLATION

Název balíčku je složen ze slov uncertainty a interpolation, což v českém překladu znamená nejistota a interpolace. Oficiální zkrácený název pro R pak zní UncerIn, pod touto zkratkou bude figurovat v prostředí R. Balíček byl vytvořen pro praktické vyzkoušení funkčnosti bakalářské práce. Hlavním cílem bylo ověření platnosti, zda je možné, propojení a zprovoznění právě modelů nejistoty a interpolačních procesů. Tato hypotéza byla potvrzena a přijata. V rámci práce tedy nebylo plánováno zabývat se parametrizací, či jinými detailnějšími principy zvolených funkcí v rámci interpolačních procesů. Z tohoto důvodu je zde poměrně velký prostor pro případné rozšiřování celého projektu.

3.1 Třídy dat

Interpolace v R vyžadují jistý vytyčený vstup dat, samotný proces výpočtů je také pevně daný. Je zde nutný vstup několika parametrů, bez kterých není možné dosáhnout správného výsledku. Těmito vstupy jsou např. souřadnice, grid, interpolované proměnné či další vybrané parametry funkcí.

Vzhledem k jisté míře automatizace všech funkcí a nastudovaným faktům, bylo přistoupeno k vytvoření vlastního typu dat o známém formátu a v objektovém návrhu verze *S3*. V současnosti existují dva tipy objektů, a to *S3* a *S4*, přičemž první z nich je vývojově starším a dnes stále dominujícím tipem v prostředí R (Matloff, 2011). Celý objekt pak dostává známou, později vždy na vstupu testovanou charakteristickou třídu, která zajišťuje správnost dat.

Programový kód 1: Zabalení a určení třídy dat (z funkce spatialPoints)

```
colums = c("x", "y", "z")
data = list(x = data[, colums[1]], y = data[, colums[2]], z = data
        [, colums[3]])
class(data) <- {"spatialPoints"}
```

Výše je uvedena část kódu pro převod objektu do třídy *spatialPoints*, k čemuž byla vytvořena stejnojmenná funkce spatialPoints. Název této funkce v sobě skrývá pojem prostorové body, čímž naznačuje, že v dalších krocích budeme pokračovat v prostoru a je důležité, aby to bylo zřejmé i ze samotné třídy objektu. Tento kód zabaluje data, pomocí příkazu list, do podoby souřadnic *x, y, z* a přidělí jim vybraný typ třídy. Vzhledem k faktu, že pro navázání dalších kroků, je vyžadována třída prvků *spatialPoints*, tak je zde uvedena na prvním místě a je preferována. Avšak, nelze opomenout i další důležité možnosti, které mohou být při práci žádány. Z tohoto důvodu je, díky dalším přiloženým funkcím, možné vytvářet i třídy typu *dataframe* či *spatialPointsDataFrame*. V tomto kroku je tedy dostatečně pojištěna správnost vstupu dat s využitím očekávané třídy dat. Nespornou výhodou navíc je, že tuto funkci lze aplikovat na jakákoli data o správném formátu, tedy matici o sloupcích x, y, z. Uživatelé si tak mohou své data libovolně převádět do správné formy pro další procesy.

Data dostala tři primární atributy *x, y, z,* kde první dva reprezentují souřadnice a třetí je zkoumaná proměnná. Zároveň jsou generované hodnoty uspořádávány do požadovaných sloupců.

Avšak *spatialPoints* nebyla jedinou třídou, která byla vytvořena. Po implementaci modelů nejistoty data dostávají dva nové sloupce, tím rozšiřují objekt a ten poté již nevyhovuje

dosavadní třídě. Z tohoto důvodu byla přidána ještě druhá funkce, která převádí objekt do nové třídy *uncertainSpatialPoints*. Na vstupu jsou zde očekávány dva možné formáty dat. Prvním formátem je matice, kde je očekávána jistá struktura sloupců. Pakliže by sloupce nebyly vyhovující, tak bude proces ukončen a uživateli se zobrazí chybové hlášení. V druhém případě, kdy na vstupu funkce nebude matice, ale jakýkoli jiný formát, je řešení mnohem flexibilnější. Stačí na vstupu funkce definovat jednotlivé sloupce, které musí být stejné délky. Tím je podchycen a vyřešen vstup různých formátů dat, což řada uživatelů jistě ocení.

Programový kód 2: Formáty vstupních dat (z funkce uncertainSpatialPoints)

```
uncertainSpatialPoints.matrix <- function (data)
uncertainSpatialPoints.default <- function (x = NULL, y = NULL,
uncertaintyLower = NULL, modalValue = NULL, uncertaintyUpper
= NULL)
```

Podobně jako spatialPoints, i tato funkce převádí data do žádaných formátů s tím rozdílem, že upravuje názvy atributů a kontroluje jejich existenci. Atributy, po propojení kódu modelu a této konverze dat, pak nabývají názvů pro souřadnice *x, y* a jednotlivé hodnoty nejistoty jsou pojmenovány jako *uncertaintyLower, modalValue* a *uncertaintyUpper*. Anglické názvy výsledků nejistoty představují hodnoty dolní hranice, střední hodnoty a horní hranice. Výsledkem jsou poté opět data ve vyhovujícím formátu, který splňuje veškeré předpoklady pro další využití.

3.2 Generování testovacích dat

Tvorba dat byla založena na myšlenkách potřeb pro získání tří primárních atributů bodů. Tyto atributy pak představují souřadnice *x, y* a hodnota zkoumané proměnné *z*. Vytvořené hodnoty poté byly využity jako testovací data výsledného balíčku.

3.3 Modely nejistoty

Nyní, když máme základní data, můžeme přistoupit ke tvorbě modelů nejistoty. Zjednodušeně lze tvrdit, že se jedná o modifikaci zkoumané proměnné uvnitř dat (základního atributu *z*). Pro demonstraci tvorby modelů nejistoty v R lze uvést například model zakládající se na procentuální nepřesnosti. Představme si vzorovou situaci, kdy nejmenovaná společnost získá kontrakt ke zpracování dat o určitém území. Zkreslení výsledných dat, vlivem různých chyb při sběru, se může pohybovat řekněme na hranici 3 procent oproti skutečnosti. Podobná myšlenka byla i inspirací pro tento model nejistoty. Čili na vstupu funkce jsou vstupní data (naměřeně hodnoty) a zmíněná hodnota procentuální nepřesnosti dat (počet procent). Nejprve byla určena chybová odchylka od původních dat. Poté byla tato množina hodnot převedena do pomocné proměnné *modify*. Výsledky pak musely být ještě přepočítány do absolutních hodnot, čímž se předešlo případné chybě v dalších výpočetních procesech. Touto chybou je myšlena kolize při matematickém odčítání hodnot, kde muže nastat nežádaná situace odečítání záporného čísla, jelikož z procesu odčítání se stane sčítání. Pro vytvoření samotného modelu nejistoty pak již jen stačilo, v případě určení dolní hranice, odečíst tuto proměnnou od zkoumané množiny vstupních dat. V opačném případě, tedy určení horní hranice, byly ke vstupním datům tyto hodnoty přičteny. Na konci funkce dostaneme zpět vstupní data a k nim přidané hraniční nepřesnosti v rámci zadaného rozpětí. Stejným postupem pak byly vytvořeny i ostatní funkce pro vytváření nejistoty. Tyto vybrané metody byly založeny na modifikaci zkoumané proměnné pomocí konstantní hodnoty, korelované chyby, náhodné procentuální či numerické nepřesnosti v rámci intervalu a také za pomoci náhodné odchylky.

Programový kód 3: Výpočet nejistoty na základě procentuální nepřesnosti (z funkce uncertaintyPercent)

```
percent = (data$z/100) * numberOfPercent
modify = abs(percent)
uncertaintyLower = data$z - modify
uncertaintyUpper = data$z + modify
```

V rámci bakalářské práce bylo vytvořeno celkem 6 funkcí, tedy 6 různých modelů, které nejistotu na vstupních datech vytváří. Všechny názvy kódů byly navrženy tak, aby vždy začínaly pojmem uncertainty (nejistota) a pokračovaly vystihujícím výrazem, který charakterizoval onu výslednou nejistotu.

1) uncertaintyConstant,
2) uncertaintyError,
3) uncertaintyPercent,
4) uncertaintyRandomPercent,
5) uncertaintyRandomNumber,
6) uncertaintyRandomDeviate.

Jak je již z názvů patrné, jsou jednotlivé modely založeny na matematické kalkulaci. Z hlediska principu průběhu se od sebe příliš neliší. Jediným razantním rozdílem je samotná kalkulace výsledného modelu, která probíhá podle zvoleného typu nejistoty. Na vstupu funkcí jsou vždy data a parametry pro výpočet nejistoty. Data budou na počátku ihned zkontrolována, zda jsou vyhovující. Tato kontrola je však vždy dodržena díky předešlé editaci dat do očekávaného formátu. Posléze následuje průběh tvorby modelu nejistoty a na závěr jsou veškeré výsledky zabaleny a vráceny uživateli v objektu třídy *uncertainSpatialPoints*. Tento typ třídy je logicky dále očekáván na vstupu pro interpolace.

3.4 Grid

Samotný princip použitých interpolačních funkcí je pevně daný. Kromě vstupních dat a jejich parametrů, vyžadují na vstupu i grid. Zvolený grid poté slouží jako mřížka pro výsledné hodnoty interpolace. Samozřejmostí tedy je, že se ve výsledném balíčku bude vyskytovat i patřičná funkce pro generování vlastních gridů uživatele.

3.5 Interpolace

Otázka interpolačních algoritmů byla vyřešena formou tří funkcí a jednou nástavbou. Tyto funkce reprezentují tři druhy interpolací, a to metody IDW, spline a kriging. Nástavbu pak představuje kód pro odhad variogramu, který vstupuje do procesu krigingu. Prakticky vrací uživateli automaticky vybraný model variogramu, včetně jeho definujících hodnot range, sill a nugget.

Protože úkolem práce nebylo zabývat se parametrizací nebo detailnějšími principy interpolačních funkcí, postačily na vstupu pouze 2 parametry pro úspěšný chod celé interpolační procedury. Těmi byly vstupní data a zvolený grid. Parametrizace interpolací byla převzata z přidružených originálních, tedy původních, interpolačních funkcí. Po spuštění procesu budou nejprve otestována vybraná vstupní data, zda jsou ve správném požadovaném formátu. Poté přichází na řadu úpravy dat do takové podoby, aby mohla být aplikována do interpolací.

Programový kód 4: Zvolená krigovací funkce a její parametry(z funkce kriging)

```
autoKrige(uncertaintyLower ~ x + y, data_frame, grid_frame)
autoKrige(modalValue ~ x + y, data_frame, grid_frame)
autoKrige(uncertaintyUpper ~ x + y, data_frame, grid_frame)
```

Jádrem jsou pak vybrané interpolační funkce do kterých vstupují potřebné parametry a zkoumané proměnné. Zmíněné proměnné jsou samozřejmě střední hodnota a její dolní, horní hranice vypočítané na základě modelu nejistoty. Dalším parametrem jsou souřadnice, které však nebylo problém získat ze vstupních dat. Předešlé hodnoty byly tedy získány z datového objektu a k nim byl navíc přidán zvolený grid, který zkompletoval základní potřebné argumenty pro interpolační funkce. Ze získaných výsledků byly vybrány pouze partie obsahující vypočítané predikce, tedy zájmové výsledky interpolací. Na závěr bylo vše opět zabaleno a převedeno do pohodlného formátu pro případné další využití.

Reference

MATLOFF, N. THE ART OF R PROGRAMMING A Tour of Statistical Software Design. William Pollock, 2011.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. Dostupné z: http: //www.R-project.org.