A generalization of inverse distance weighting and an equivalence relationship to noise-free Gaussian process interpolation via Riesz representation theorem

DOI: 10.1080/03081087.2017.1337057
Handle: http://hdl.handle.net/1942/26266

A generalization of inverse distance weighting and an equivalence relationship to noise-free Gaussian process interpolation via Riesz representation theorem

Wim De Mulder*[1], Geert Molenberghs[2,1], and Geert Verbeke[1,2]

[1]I-BioStat, KU Leuven, Leuven, Belgium
[2]I-BioStat, Universiteit Hasselt, Hasselt, Belgium

January 31, 2017

Abstract

In this paper we show the relationship between two seemingly unrelated approximation techniques. On the one hand a certain class of Gaussian process based interpolation methods, and on the other hand inverse distance weighting, which has been developed in the context of spatial analysis where there is often a need for interpolating from irregularly-spaced data to produce a continuous surface. We develop a generalization of inverse distance weighting and show that it is equivalent to the approximation provided by the class of Gaussian process based interpolation methods. The equivalence is established via an elegant application of Riesz representation theorem concerning the dual of a Hilbert space. It is thus demonstrated how a classical theorem in linear algebra connects two disparate domains.

Keywords— Riesz representation theorem; Gaussian process; inverse distance weighting; interpolation; kriging

1 Introduction

This paper contains a theoretical contribution to some techniques that interpolate given observations. In particular, we establish an interesting relationship between inverse distance weighting (IDW) and some Gaussian process (GP) based interpolation techniques. IDW is a rather intuitive interpolation method in a metric space setting, originally developed by Shepard in the

*Corresponding author. Email address: wim.demulder@cs.kuleuven.be

context of spatial analysis and geographic information systems [15]. It is still applied in many practical approximation problems (see, e.g., [2], [3], [4], [6], [7]). We refer to Section 4 for a succinct formulation of IDW. A more advanced class of interpolation techniques is based on Gaussian processes, which are statistical models where every point in some continuous input space has an associated output that is conceived as a normally distributed random variable. Several fields, such as machine learning and emulation, make use of Gaussian processes for certain approximation tasks, and the exact model used for the task can be somewhat different, depending on the precise goal and the underlying assumptions (e.g. univariate vs. multivariate data, observations generated by a deterministic function vs. generated by a stochastic process, ...).

Our contribution is the formulation of a more general version of the originally developed IDW technique, and then proving the mathematical equivalence of this formulation to certain Gaussian process based interpolation techniques that are constructed in a noise-free environment. The term noise-free environment is used here in the sense that the given observations are free of measurement noise. This may seem a severe assumption, but it should be stressed that the goal of this paper is not to develop a realistic model to be applied to physical measurements. Instead, our purpose is to show how seemingly unrelated and independently developed techniques, i.e. IDW and certain GP based interpolation techniques, are connected to each other. The fact that IDW has been developed to apply to noise-free observations explains at once our restriction to noise-free GP based interpolation methods. Furthermore, it is worth mentioning that there are applications where the assumption of absence of noise is acceptable. One large category of such applications is the emulation of expensive deterministic computer simulation experiments, where the objective is to obtain a fast-running approximation for a given complex, time-consuming deterministic simulator [8]. The fact that in this case the numerical observations are generated by a computer and that the underlying function is supposedly deterministic together imply that noise can safely be ignored.

The paper is structured as follows. Section 2 provides some references to research that considers some connections between other approximation methods, just meant as interesting background material. The noise-free GP interpolation that is the focus of this work is outlined in Section 3. Inverse distance weighting, as it was originally developed by Shepard, is outlined in Section 4. The same section discusses a generalization of it, together with the properties of this more general approximation method. In Section 5 we examine a variation on inverse distance weighting by taking into account a prior approximation method. Both inverse distance weighting and the variation on it approximate an unknown value via a convex combination of known values. The requirement of convexity is dropped in Section 6. It is shown how this results in the relationship with noise-free GP interpolation.

2 Related work

Our work extends the amount of connections that have already been established between existing approximation methods. Without intending to be exhaustive, we list some examples of such connections. First, it can be shown that spline and generalized spline smoothing is equivalent to Bayesian estimation with a partially improper prior [18]. The authors interpret this result as saying that spline smoothing is a natural solution to the regression problem when one is given a set of regression functions but one also wants to hedge against the possibility that the true model is not exactly in the span of the given regression functions. Secondly, Neal has shown a connection between Gaussian processes and artificial neural networks [13]. His connection states that the properties of a neural network with one hidden layer converge to those of a Gaussian process as the number of hidden neurons tends to infinity if standard weight decay priors are assumed. This has resulted in the question whether supervised neural networks should be dismissed in favor of Gaussian processes [12]. A third and interesting relationship is between Gaussian processes and the Kalman filter [11, 14]. This connection has resulted in hybrid computationally efficient methods, such as K-nearest neighbor based Kalman filter Gaussian process (KNN-KFGP) regression, a regression method that circumvents some of the computational deficiencies of Gaussian processes when the data set is large or spatially nonstationary [19].

3 Noise-free GP interpolation

The noise-free case of Gaussian process regression, also called kriging model [5, 16], and Gaussian process emulation [10] have similar formulations. Gaussian process emulation is a subclass of surrogate modeling [8], where the objective is to obtain a fast-running approximation for a complex, time-consuming model. The surrogate model in Gaussian process emulation is conveniently called the emulator and is intended to approximate a deterministic, possibly unknown, function ν. Kriging originated in geostatistics as a method to perform predictions, given a set of observations. Several versions of GP emulation and kriging have been developed, and in this paper we restrict to a noise-free formulation that interpolates the observations. To describe this formulation we will rely on the initially developed GP emulation framework [10] and on the Bayesian approach to kriging [9, 17]. We will refer to this formulation as noise-free GP interpolation. To establish the connection with IDW, which is a non statistical method, we will not pay much attention to the distributional aspect of the involved Gaussian process. The concept of interest in this paper is the posterior mean that results from a Bayesian analysis and that is used to calculate expected values for

the random variable in output space associated to some given point in input space. Some more detail is provided below. We will refer to this posterior mean as the emulator. Furthermore, we restrict attention to the case where $\nu(\mathbf{z}) \in \mathbb{R}$, for any given input vector $\mathbf{z} \in \mathbb{R}^p$ for some $p \in \mathbb{N}$. This output $\nu(\mathbf{z})$ is considered a realization of a random variable $\zeta(\mathbf{z})$.

The construction of the emulator requires pairs of the form $(\mathbf{z}_i, \nu(\mathbf{z}_i))$, obtained by applying ν to a limited number of input points $\mathbf{z}_1, \ldots, \mathbf{z}_n$. We call the set of input points to which ν has been applied the training data set and denote it as $\top = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$, where it is assumed that $\mathbf{z}_i \neq \mathbf{z}_j$ if $i \neq j$. The corresponding vector of outputs is denoted as $\nu(\top) = [\nu(\mathbf{z}_1), \ldots, \nu(\mathbf{z}_n)]^T$. Sections 3.1-3.3 describe the emulator in several stages, starting with a description of the prior mean that can be considered a first approximation to the output of ν in a given input point, followed by the introduction of a more general class of correlation functions that are often used in GP applications, and ending with a description of the posterior mean (also called the emulator in this paper) that is used as an improvement to approximations provided by the prior mean.

3.1 Prior mean

The determination of the posterior mean consists of two steps. First, before training data has been obtained, a *prior* mean $m(\mathbf{z})$ is considered, which is modeled as a linear combination of user-chosen regression functions applied to a given input $\mathbf{z}$. That is

$$E[\zeta(\mathbf{z}) \,|\, \beta] \;=\; \sum_{i=1}^{q} \beta_i \, h_i(\mathbf{z}) \tag{1}$$

with h_i the regression functions and with $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_q]^T \in \mathbb{R}^q$ the coefficients. The value of $\boldsymbol{\beta}$ is immaterial in our discussion and thus we consider $\boldsymbol{\beta}$ an arbitrary vector in Euclidean space.

Definition 3.1. *We define the following matrix, given input vectors* $\mathbf{x}_1, \ldots, \mathbf{x}_l$*:*

$$H(\mathbf{x}_1, \ldots, \mathbf{x}_l) \;=\; \begin{bmatrix} h_1(\mathbf{x}_1) & \ldots & h_q(\mathbf{x}_1) \\ \ldots & & \\ h_1(\mathbf{x}_l) & \ldots & h_q(\mathbf{x}_l) \end{bmatrix}$$

With this notation the prior mean can be written in short as

$$m(\mathbf{z}) \;=\; H(\mathbf{z})\,\boldsymbol{\beta} \tag{2}$$

Definition 3.2. *We introduce the shorthand notation* $H = H(\mathbf{z}_1, \ldots, \mathbf{z}_n)$.

3.2 Introduction of a general class of correlation functions

The correlation between two given random variables $\zeta(\mathbf{z})$ and $\zeta(\mathbf{z}')$ is modeled via a user-chosen correlation function $c(\mathbf{z}, \mathbf{z}')$. We introduce here a general class of correlation functions:

$$c(\mathbf{z}, \mathbf{z}') \quad = \quad \Gamma(d(\mathbf{z}, \mathbf{z}')) \tag{3}$$

where Γ is as in the following definition and where d is a metric.

Definition 3.3. Γ *is a function on the nonnegative real line with the following properties:*

1. $0 \leq \Gamma(x) \leq 1, \forall x \geq 0$
2. Γ *is non-increasing*
3. Γ *is continuous*
4. $\Gamma(0) = 1$

As an example, the widely adopted Gaussian correlation function

$$c_g(\mathbf{z}, \mathbf{z}') \quad = \quad \exp\Big(-(\mathbf{z} - \mathbf{z}')^T M(\mathbf{z} - \mathbf{z}')\Big) \tag{4}$$

with M a positive-definite matrix, is a member of this more general class of correlation functions. Indeed, let $\Gamma(x) = \exp(-x^2)$ such that this Γ has the properties required by definition 3.3. It is then seen that $c_g(\mathbf{z}, \mathbf{z}') = \Gamma(d_g(\mathbf{z}, \mathbf{z}'))$ with

$$d_g(\mathbf{z}, \mathbf{z}') \quad = \quad \sqrt{(\mathbf{z} - \mathbf{z}')^T M(\mathbf{z} - \mathbf{z}')}.$$

3.3 Posterior mean

Definition 3.4. *The matrix A contains the correlations between the output random variables corresponding to the training data set, i.e. $A(i,j) = c(\mathbf{z}_i, \mathbf{z}_j)$, where $A(i,j)$ denotes the element on the ith row and jth column of A.*

Definition 3.5.

$$U(\mathbf{z}) \quad = \quad [c(\mathbf{z}, \mathbf{z}_1), \ldots, c(\mathbf{z}, \mathbf{z}_n)]^T$$

for an arbitrary input point $\mathbf{z}$.

Definition 3.6. *We define the following* error vector $\mathbf{e}$*:*

$$\mathbf{e} \quad = \quad \nu(\top) - H\boldsymbol{\beta} \tag{5}$$

The vector $\mathbf{e}$ is an error vector in the sense that it contains the differences between known, correct output values in the training data points and the approximations of these output values via the prior mean.
The posterior mean $m^\star(\mathbf{z})$ in any input point $\mathbf{z}$ is then given by [10]

$$m^\star(\mathbf{z}) \quad = \quad H(\mathbf{z})\boldsymbol{\beta} + U^T(\mathbf{z})\, A^{-1}\, \mathbf{e} \tag{6}$$

The quantity $m^\star(\mathbf{z})$ approximates or predicts the value of ν in $\mathbf{z}$ in noise-free GP interpolation.

4 Inverse distance weighting

4.1 Introduction

IDW approximates the unknown value $\nu(\mathbf{z})$ in a given point $\mathbf{z}$ as a weighted average of the known values in the training data points $\top = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$, where each weight decreases with increasing distance to $\mathbf{z}$.
The IDW method as originally proposed by Shepard is given by [15]:

$$\hat{\nu}(\mathbf{z}) \quad = \quad \sum_{i=1}^{n} \frac{w_i(\mathbf{z})}{\sum_{j=1}^{n} w_j(\mathbf{z})} \nu(\mathbf{z}_i) \qquad \text{if } d(\mathbf{z}, \mathbf{z}_i) \neq 0 \text{ for all } i \tag{7}$$

$$\quad = \quad \nu(\mathbf{z}_i) \qquad \text{if } d(\mathbf{z}, \mathbf{z}_i) = 0 \text{ for some } i \tag{8}$$

where

$$w_i(\mathbf{z}) \quad = \quad \frac{1}{d(\mathbf{z}, \mathbf{z}_i)^\alpha} \tag{9}$$

where d is any metric and where α is a constant larger than zero. From (8) it is clear that this method interpolates the values in the training data points, i.e. $\hat{\nu}(\mathbf{z}) = \nu(\mathbf{z}), \forall \mathbf{z} \in \top$.

4.2 Generalization

We propose the following generalization of the weights (9):

$$w_i(\mathbf{z}) \quad = \quad F(d(\mathbf{z}, \mathbf{z}_i)) \tag{10}$$

where F is defined on the positive real line and has the following properties:

1. $F(x) \geq 0, \forall x > 0$
2. F is non-increasing
3. F is continuous
4. $\lim_{x \to 0, x > 0} F(x) = +\infty$

The first property ensures that $\hat{\nu}(\mathbf{z})$ is a convex combination of the true values $\nu(\mathbf{z}_i)$, a property possessed by the IDW method (7)-(9). The second property ensures a fundamental characteristic of IDW, namely that the influence of a certain training data point on the determination of $\nu(\mathbf{z})$ diminishes with increasing distance between $\mathbf{z}$ and that training data point. The last two properties will be used to establish continuity of $\hat{\nu}$ in section 4.3 below. Continuity of $\hat{\nu}$ might also be considered an essential and desirable feature of IDW.

For example, $F(x) = 1/x^{\alpha}$ for $x > 0$ and with $\alpha > 0$ fulfills the above properties, showing that (10) is indeed a generalization of (9).

4.3 Continuity of $\hat{\nu}$

Theorem 4.1. *$\hat{\nu}(\mathbf{z})$ given by (7)-(8) with $w_i(\mathbf{z})$ given by (10) is continuous in $\mathbf{z}$, where continuity is defined with respect to d.*

Proof. Consider a given $\mathbf{z} \in \mathbb{R}^p$ and consider any sequence $\mathbf{x}_m \to \mathbf{z}$, which means that $d(\mathbf{x}_m, \mathbf{z})$ goes to zero as m goes to infinity. The triangle inequality then implies that $|d(\mathbf{x}_m, \mathbf{z}_i) - d(\mathbf{z}, \mathbf{z}_i)| \leq d(\mathbf{x}_m, \mathbf{z})$ and thus $d(\mathbf{x}_m, \mathbf{z}_i) \to d(\mathbf{z}, \mathbf{z}_i)$. From the continuity of F it then follows that $F(d(\mathbf{x}_m, \mathbf{z}_i)) \to F(d(\mathbf{z}, \mathbf{z}_i))$. By (10) this is equivalent to stating that $w_i(\mathbf{x}_m) \to w_i(\mathbf{z})$.
We now consider two cases. First, let $\mathbf{z} \notin \top$. From $w_i(\mathbf{x}_m) \to w_i(\mathbf{z})$ and (7) we then deduce that

$$\hat{\nu}(\mathbf{x}_m) = \sum_{i=1}^{n} \frac{w_i(\mathbf{x}_m)}{\sum_{j=1}^{n} w_j(\mathbf{x}_m)} \nu(\mathbf{z}_i) \quad \to \quad \sum_{i=1}^{n} \frac{w_i(\mathbf{z})}{\sum_{j=1}^{n} w_j(\mathbf{z})} \nu(\mathbf{z}_i) = \hat{\nu}(\mathbf{z})$$

showing the continuity of $\hat{\nu}$ in the non-training data points.
Secondly, suppose that $\mathbf{z} = \mathbf{z}_k \in \top$. The properties $\lim_{x \to 0, x > 0} F(x) = +\infty$ and $\mathbf{z}_i \neq \mathbf{z}_j$ if $i \neq j$ imply that

$$\frac{w_k(\mathbf{x}_m)}{\sum_{j=1}^{n} w_j(\mathbf{x}_m)} \to 1 \qquad \text{and} \qquad \frac{w_i(\mathbf{x}_m)}{\sum_{j=1}^{n} w_j(\mathbf{x}_m)} \to 0, \quad i \neq k$$

and thus $\hat{\nu}(\mathbf{x}_m) \to \nu(\mathbf{z}_k) = \hat{\nu}(\mathbf{z}_k)$, thereby making use of (8).
Thus $\mathbf{x}_m \to \mathbf{z}$ implies that $\hat{\nu}(\mathbf{x}_m) \to \hat{\nu}(\mathbf{z})$ whether $\mathbf{z} \in \top$ or $\mathbf{z} \notin \top$. □

Definition 4.2. *For $\mathbf{z} \notin \top$:*

$$W(\mathbf{z}) \quad = \quad \Big[\frac{w_1(\mathbf{z})}{\sum_{j=1}^{n} w_j(\mathbf{z})}, \ldots, \frac{w_n(\mathbf{z})}{\sum_{j=1}^{n} w_j(\mathbf{z})} \Big]^T$$

With this definition (7) can be rewritten using the Euclidean inner product $< ., . >$, such that an equivalent way to describe (7)-(8) is:

$$\begin{aligned} \hat{\nu}(\mathbf{z}) &= < W(\mathbf{z}), \nu(\top) > && \text{if } d(\mathbf{z}, \mathbf{z}_i) \neq 0 \text{ for all } i \qquad (11) \\ &= \nu(\mathbf{z}_i) && \text{if } d(\mathbf{z}, \mathbf{z}_i) = 0 \text{ for some } i \qquad (12) \end{aligned}$$

Theorem 4.3. *$\hat{\nu}(\mathbf{z})$ given by (11)-(12) with $w_i(\mathbf{z})$ given by (10) is continuous in $W(\mathbf{z})$ and in $\nu(\top)$ where continuity is defined in terms of the Euclidean norm.*

Proof. We first proof continuity in $W(\mathbf{z})$. Let $\mathbf{z} \in \mathbb{R}^p$ and consider any sequence $W_m \to W(\mathbf{z})$. Thus $\Big|\Big|W_m - W(\mathbf{z})\Big|\Big| \to 0$ as $m \to +\infty$.
First suppose that $\mathbf{z} \notin \top$. Using the Cauchy-Schwarz inequality, it follows that:

$$\begin{aligned} |< W_m, \nu(\top) > - < W(\mathbf{z}), \nu(\top) >| &= |< W_m - W(\mathbf{z}), \nu(\top) >| \\ &\leq \Big|\Big|W_m - W(\mathbf{z})\Big|\Big| \ \Big|\Big|\nu(\top)\Big|\Big| \end{aligned}$$

This shows that $< W_m, \nu(\top) > \to < W(\mathbf{z}), \nu(\top) >$ as m goes to infinity. By (11) this is equivalent to $< W_m, \nu(\top) > \to \hat{\nu}(\mathbf{z})$, showing continuity in $W(\mathbf{z})$ when $\mathbf{z} \notin \top$.
Now let $\mathbf{z} = \mathbf{z}_k \in \top$. The proof of theorem 4.1 tells us that $W_m \to \boldsymbol{\xi}_k$, with $\boldsymbol{\xi}_k$ the kth Euclidean standard vector. This implies that $< W_m, \nu(\top) > \to < \boldsymbol{\xi}_k, \nu(\top) >= \nu(\mathbf{z}_k) = \hat{\nu}(\mathbf{z}_k)$, using (12). Continuity in $W(\mathbf{z})$ is then established for all input points $\mathbf{z}$.

We now proof continuity in $\nu(\top)$. Consider any sequence $\nu_m \to \nu(\top)$ and any $\mathbf{z} \notin \top$. As in the first part of the proof we then have that $< W(\mathbf{z}), \nu_m > \to < W(\mathbf{z}), \nu(\top) >= \hat{\nu}(\mathbf{z})$.
Now let $\mathbf{z} = \mathbf{z}_k \in \top$. Irrespective of the values that the vector ν_m contains, the value $\hat{\nu}(\mathbf{z}_k)$ is given by $\nu(\mathbf{z}_k)$ because of (12). The sequence of interest corresponding to ν_m is thus the constant sequence $\hat{\nu}(\mathbf{z}_k)$ which evidently converges to $\hat{\nu}(\mathbf{z}_k)$. □

4.4 Essential properties of $\hat{\nu}$ in IDW

The above considerations show that $\hat{\nu}$ has the following fundamental properties:

1. $\hat{\nu}(\mathbf{z})$ determines an approximation for $\nu(\mathbf{z})$ in terms of $W(\mathbf{z})$, a vector where each component is a weight that decreases with increasing distance between $\mathbf{z}$ and the corresponding training data point, as well as in terms of $\nu(\top)$, a vector where each component is the true value of ν in a training data point.
2. $\hat{\nu}(\mathbf{z})$ is continuous in $\mathbf{z}$ with respect to d.
3. $\hat{\nu}(\mathbf{z})$ is continuous in $W(\mathbf{z})$ and in $\nu(\top)$ with respect to Euclidean distance.
4. If $\mathbf{z} \notin \top$, $\hat{\nu}(\mathbf{z})$ is linear in $W(\mathbf{z})$ and in $\nu(\top)$.
5. $\hat{\nu}(\mathbf{z}_k) = \nu(\mathbf{z}_k)$ for all $\mathbf{z}_k \in \top$.

5 Error-based inverse distance weighting

In this section we propose a variation on the inverse distance weighting method, which we call error-based inverse distance weighting (EIDW). The variation is developed in two steps. First, section 5.1 presents the main modification to IDW by replacing the weighted average of true output values by the value of a given prior approximation method corrected by a weighted average of error values. The second step, discussed in section 5.2, is merely considered to increase elegance by replacing the two expressions that together describe IDW, given by (7)-(8) or the equivalent description given by (11)-(12), by a single expression.

5.1 First modification

Instead of defining approximations in terms of a weighted average of true values, we may use weighted averages of *error values* to correct the value determined by another approximation method, which we call the prior approximation method ρ. To be more precise, let the prior approximation method be given by a linear combination of user-chosen regression functions with already determined coefficients. That is, we choose it as the prior mean in Gaussian process emulation: $\rho(\mathbf{z}) = H(\mathbf{z})\boldsymbol{\beta}$, as given by (2). The vector $\mathbf{e} = [e_1, \ldots, e_n]^T$, defined by (5), then contains the errors between the correct output values in the training points and their approximations by the prior approximation method. A variation on IDW is then to determine $\hat{\nu}(\mathbf{z})$ as the value determined by the prior approximation method corrected with a weighted average of the error values e_k, where the weight increases as $d(\mathbf{z}, \mathbf{z}_k)$ decreases:

$$\begin{aligned} \hat{\nu}(\mathbf{z}) &= H(\mathbf{z})\boldsymbol{\beta} + < W(\mathbf{z}), \mathbf{e} > && \text{if } d(\mathbf{z}, \mathbf{z}_i) \neq 0 \text{ for all } i \quad (13) \\ &= H(\mathbf{z})\boldsymbol{\beta} + e_i && \text{if } d(\mathbf{z}, \mathbf{z}_i) = 0 \text{ for some } i \quad (14) \end{aligned}$$

The strong similarity with (11)-(12) is obvious.

5.2 Second modification

We return to IDW described in section 4. It would be more elegant if the two expressions (11)-(12) that define IDW could be combined into a single expression.
One intuitive idea to accomplish this is to allow that the expression (11) for $\hat{\nu}(\mathbf{z})$ in *non-training* data points $\mathbf{z}$ is applicable to *all* data points. However, this is prevented by definition 4.2, where $W(\mathbf{z})$ is only defined for $\mathbf{z} \notin \top$. The reason for this limitation is that $w_i(\mathbf{z}_i)$ is not necessarily defined for $i \in \{1, \ldots, n\}$. Indeed, for the original IDW method developed by Shepard we see from (9) that $\lim_{\mathbf{z} \to \mathbf{z}_i} w_i(\mathbf{z}) = +\infty$. This property was retained when, in section 4.2, we generalized $w_i(\mathbf{z})$ in Shepard's method to $w_i(\mathbf{z}) = F(d(\mathbf{z}, \mathbf{z}_i))$

by imposing that $\lim_{x\to 0,x>0} F(x) = +\infty$. We used this property in showing that $\hat{\nu}$ is continuous in the training data points, see theorem 4.1.
A closer look at the proof of theorem 4.1 reveals that this specific property of F was used to deduce that if $\mathbf{x}_m \to \mathbf{z}_i$ then $W(\mathbf{x}_m) \to \boldsymbol{\xi}_i$, where $\boldsymbol{\xi}_i$ denotes the ith Euclidean standard vector. Continuity in $\mathbf{z}_i \in \top$ was then implied by this assertion. Consequently, the property $\lim_{x\to 0,x>0} F(x) = +\infty$ is not needed if we extend the definition of $W(\mathbf{z})$ to all $\mathbf{z}$ as follows:

$$W(\mathbf{z}) = \Big[\frac{w_1(\mathbf{z})}{\sum_{j=1}^n w_j(\mathbf{z})}, \ldots, \frac{w_n(\mathbf{z})}{\sum_{j=1}^n w_j(\mathbf{z})}\Big]^T \quad \text{if } \mathbf{z} \notin \top \tag{15}$$

$$= \boldsymbol{\xi}_i \quad \text{if } \mathbf{z} = \mathbf{z}_i \in \top \tag{16}$$

With this definition, $\hat{\nu} =< W(\mathbf{z}), \nu(\top) >$ is well defined for all $\mathbf{z}$ and not just for the non-training data points as in (11) where $W(\mathbf{z})$ is given by definition 4.2.

It might be objected that no increase in elegance has been acquired, as the single expression for $\hat{\nu}$ is obtained by introducing an extra expression for $W(\mathbf{z})$. This objection is completely justified. As a next step, we observe that (15)-(16) can equivalently be expressed as the vector

$$\tilde{W}(\mathbf{z}) = [\tilde{w}_1(\mathbf{z}), \ldots, \tilde{w}_n(\mathbf{z})]^T \tag{17}$$

with $0 \leq \tilde{w}_i(\mathbf{z}) \leq 1$, $\sum_i \tilde{w}_i(\mathbf{z}) = 1$ and $\tilde{w}_i(\mathbf{z}_k) = \delta_{ik}$, where $\delta_{ik} = 1$ if $i = k$ and 0 otherwise. The components $\tilde{w}_i(\mathbf{z})$ of $\tilde{W}(\mathbf{z})$ can then still be defined as

$$\tilde{w}_i(\mathbf{z}) = F(d(\mathbf{z}, \mathbf{z}_i)) \tag{18}$$

in the same way as we defined the components of $W(\mathbf{z})$ by $w_i(\mathbf{z}) = F(d(\mathbf{z}, \mathbf{z}_i))$, see (9). The main differences between $W(\mathbf{z})$ and $\tilde{W}(\mathbf{z})$ are that we have dropped the property $\lim_{x\to 0,x>0} F(x) = +\infty$ and that we introduced the additional requirements $0 \leq \tilde{w}_i(\mathbf{z}) \leq 1$, $\sum_i \tilde{w}_i(\mathbf{z}) = 1$ and $\tilde{w}_i(\mathbf{z}_k) = \delta_{ik}$.

Instead of imposing the conditions $0 \leq \tilde{w}_i(\mathbf{z}) \leq 1$ and $\tilde{w}_i(\mathbf{z}_k) = \delta_{ik}$ we may equally well impose the following additional properties on F: $0 \leq F(x) \leq 1, \forall x \geq 0$, and $F(0) = 1$. We notice that the properties of F are now exactly these of Γ, given in definition 3.3. Thus one consequence of our second modification is that F has been changed into Γ.

5.3 Synthesis of the modifications

Combining the modifications to IDW described in sections 5.1 and 5.2 results in the following description of EIDW:

$$\hat{\nu}(\mathbf{z}) = H(\mathbf{z})\boldsymbol{\beta} + < \tilde{W}(\mathbf{z}), \mathbf{e} > \tag{19}$$

with

$$\tilde{W}(\mathbf{z}) \quad = \quad [\tilde{w}_1(\mathbf{z}), \ldots, \tilde{w}_n(\mathbf{z})]^T \tag{20}$$

where the components $\tilde{w}_i(\mathbf{z})$ of $\tilde{W}(\mathbf{z})$ fulfill

$$1. \qquad \tilde{w}_i(\mathbf{z}) = \Gamma(d(\mathbf{z}, \mathbf{z}_i)) \tag{21}$$

$$2. \qquad \sum_i \tilde{w}_i(\mathbf{z}) = 1 \tag{22}$$

and this for all input points $\mathbf{z}$.

The second modification made F equivalent to Γ. From (3) and (18) it thus follows that $\tilde{w}_i(\mathbf{z}) = c(\mathbf{z}, \mathbf{z}_i)$. From (5) and (20) it is seen that $\tilde{W}(\mathbf{z}) = U(\mathbf{z})$. An equivalent description of EIDW is thus given by

$$\hat{\nu}(\mathbf{z}) \quad = \quad H(\mathbf{z})\boldsymbol{\beta} + < U(\mathbf{z}), \mathbf{e} > \tag{23}$$

with

$$\sum_i c(\mathbf{z}, \mathbf{z}_i) \quad = \quad 1 \tag{24}$$

and this for all input points $\mathbf{z}$.

We have developed EIDW as a variation on IDW. An alternative view is to consider it a generalization of IDW, since it is easy to show that the above description reduces to (11)-(12) by choosing $h_i = 0$ for all $i \in \{1, \ldots, q\}$.

5.4 Essential properties of $\hat{\nu}$ in EIDW

Which of the properties of IDW described in section 4.4 still hold under EIDW?

Instead of the first property a variation now holds. It is the essence of EIDW that $\hat{\nu}$ is not defined anymore in terms of $W(\mathbf{z})$ and $\nu(\top)$, but in terms of $\tilde{W}(\mathbf{z})$, $\mathbf{e}$ and a prior approximation method, as seen in (19).

Provided that all regression functions h_i are continuous in $\mathbf{z}$ with respect to d, the second property still holds. The proof is completely similar to the first part of the proof of theorem 4.1.

A modified form of property 3 is still valid, namely continuity of $\hat{\nu}$ in $\tilde{W}(\mathbf{z})$ and in $\mathbf{e}$. This follows from the continuity of an inner product.

Property 4 also still holds in a modified form, namely linearity of the error term in $\tilde{W}(\mathbf{z})$, and this for all $\mathbf{z}$, and in $\mathbf{e}$. This follows from the bilinearity of an inner product.

The last property is also retained, as shown by the following theorem.

Theorem 5.1. *If $\mathbf{z}_k \in \top$, then $\hat{\nu}(\mathbf{z}_k) = \nu(\mathbf{z}_k)$, where $\hat{\nu}$ is defined by (19)-(22).*

Proof. By (19) we have that $\hat{\nu}(\mathbf{z}_k) = H(\mathbf{z}_k)\boldsymbol{\beta} + < \tilde{W}(\mathbf{z}_k), \mathbf{e} >$. Properties (21) and (22) imply that $\tilde{w}_k(\mathbf{z}_k) = \Gamma(0) = 1$, using property 4 of Γ in definition 3.3, and that $\tilde{w}_i(\mathbf{z}_k) = 0$ for $i \neq k$. In other words $\tilde{W}(\mathbf{z}_k) = \boldsymbol{\xi}_k$. Thus

$$\begin{aligned} \hat{\nu}(\mathbf{z}_k) &= H(\mathbf{z}_k)\boldsymbol{\beta} + < \boldsymbol{\xi}_k, \mathbf{e} > \\ &= H(\mathbf{z}_k)\boldsymbol{\beta} + e_k \\ &= H(\mathbf{z}_k)\boldsymbol{\beta} + \nu(\mathbf{z}_k) - H(\mathbf{z}_k)\boldsymbol{\beta} \\ &= \nu(\mathbf{z}_k) \end{aligned}$$

where we used definition 3.6 of $\mathbf{e}$. □

In summary, the main properties of $\hat{\nu}$ in EIDW are

1. $\hat{\nu}(\mathbf{z})$ determines an approximation for $\nu(\mathbf{z})$ in terms of $\tilde{W}(\mathbf{z}), \mathbf{e}$ and a prior approximation method.
2. $\hat{\nu}(\mathbf{z})$ is continuous in $\mathbf{z}$ with respect to d, provided that all h_i are continuous in $\mathbf{z}$ with respect to d.
3. $\hat{\nu}(\mathbf{z})$ is continuous in $\tilde{W}(\mathbf{z})$ and in $\mathbf{e}$ with respect to Euclidean distance.
4. The error term of $\hat{\nu}(\mathbf{z})$ is linear in $\tilde{W}(\mathbf{z})$ and in $\mathbf{e}$.
5. $\hat{\nu}(\mathbf{z}_k) = \nu(\mathbf{z}_k)$ for all $\mathbf{z}_k \in \top$.

6 Generalized error-based inverse distance weighting

6.1 Generalization of EIDW

We present a generalization of EIDW (and thus a further generalization of IDW), which we call for that reason generalized error-based inverse distance weighting (GEIDW). The generalization has to do with the requirement $\sum_i \tilde{w}_i(\mathbf{z}) = 1$, given in (22). This constraint is highly undesirable, which is motivated as follows. Let $\mathbf{z}$ and $\mathbf{z}'$ be two points of the input space with $d(\mathbf{z}', \mathbf{z}_i) > d(\mathbf{z}, \mathbf{z}_i)$ for all $i = 1, \ldots, n$. Since Γ is non-increasing it follows from (21) that $\tilde{w}_i(\mathbf{z}') \leq \tilde{w}_i(\mathbf{z}), i = 1, \ldots, n$. Due to property (22) this is only possible if $\tilde{w}_i(\mathbf{z}') = \tilde{w}_i(\mathbf{z})$ for all i. Thus although the distance to each training data point has been increased, the contribution of each error component e_i has remained constant. Having a prior approximation method at our disposal, it is preferable to take this into account by giving more confidence to this prior method for input points that are far away from the training data

points, since the large distance means that we should not expect to gain much information from the values of ν in the training data points, and vice versa. This urges us to drop constraint (22).

However, this has an unwanted side-effect, since it is then no longer guaranteed that $\hat{\nu}$ is an interpolator, which is a basic property in IDW and in EIDW. Indeed, property (22) was essential in proving theorem 5.1 on the interpolation property of $\hat{\nu}$ in EIDW.

Is it possible to modify EIDW such that the constraint (22), or equivalently property (24), is not required, without giving up the interpolation property and, preferably, without giving up the other properties of $\hat{\nu}$ in EIDW?

To this end, we generalize the correction term $< U(\mathbf{z}), \mathbf{e} >$ of $\hat{\nu}$ in (23) to $g(U(\mathbf{z}), \mathbf{e})$ where g is, for the time being, any bounded, bilinear form. Imposing these characteristics on g then already ensures that properties 1 and 4 of $\hat{\nu}$ in EIDW, described in section 5.4, are retained. Property 3 also still holds, since a bounded linear transformation is continuous. The next step is to impose further characteristics on g such that, if possible, $\hat{\nu}(\mathbf{z}) = H(\mathbf{z})\boldsymbol{\beta} + g(U(\mathbf{z}), \mathbf{e})$ is an interpolator. This gives the following theorem.

Theorem 6.1. *Given is the approximation method*

$$\hat{\nu} = H(\mathbf{z})\boldsymbol{\beta} + g(U(\mathbf{z}), \mathbf{e}) \tag{25}$$

where g is bilinear and bounded. The only method of this form that ensures $\hat{\nu}(\mathbf{z}_k) = \nu(\mathbf{z}_k), k = 1, \ldots, n$, is given by

$$\hat{\nu}(\mathbf{z}) = H(\mathbf{z})\boldsymbol{\beta} + U^T(\mathbf{z})A^{-1}\mathbf{e} \tag{26}$$

provided that A is invertible.

Proof. Consider any bounded bilinear form g. According to Riesz representation theorem for bounded sesquilinear forms on the Cartesian product of a Hilbert space with itself, there exists a unique matrix S_g, such that (25) is represented as

$$\hat{\nu}(\mathbf{z}) = H(\mathbf{z})\boldsymbol{\beta} + < S_g\, U(\mathbf{z}), \mathbf{e} >$$

The requirement $\hat{\nu}(\mathbf{z}_k), k = 1, \ldots, n$, is then equivalent to

$$\begin{aligned} H(\mathbf{z})\boldsymbol{\beta} + < S_g\, U^T(\mathbf{z}_k), \mathbf{e} > &= \nu(\mathbf{z}_k) && (27)\\ &= e_k + H(\mathbf{z})\boldsymbol{\beta} && (28) \end{aligned}$$

taking into account definition 3.6. From definitions 3.4 and 3.5 it follows that $U^T(\mathbf{z}_k) = A\boldsymbol{\xi}_k$ such that (28) becomes

$$\begin{aligned} & < S_g\, A\boldsymbol{\xi}_k, \mathbf{e} > = e_k \\ \Leftrightarrow \quad & (S_g A)(., k) = \boldsymbol{\xi}_k \end{aligned}$$

where $(S_g A)(., k)$ denotes the kth column of $S_g A$. Since this should hold for all $k \in \{1, \ldots, n\}$, it follows that $S_g A = I$. If A is invertible it follows that

$$\hat{\nu}(\mathbf{z}) \quad = \quad H(\mathbf{z})\boldsymbol{\beta} + < A^{-1} U(\mathbf{z}), \mathbf{e} > \tag{29}$$

Noticing that A^{-1} is symmetric completes the proof. □

Comparing (6) and (26), we have thus shown that generalized error-based inverse distance weighting is equivalent to noise-free GP interpolation.

One consequence of this equivalence is that it immediately follows that the posterior mean interpolates the training data points, i.e. $m^\star(\mathbf{z}_i) = \nu(\mathbf{z}_i), \forall \mathbf{z}_i \in \top$, a well known property in GP emulation [1].

We notice again that the term *generalized* error-based inverse distance weighting is appropriate, as (23) is a special case of (29), and thus of (26), with A the identity matrix. Whereas in EIDW the interpolation property is ensured by the condition $\sum_i c(\mathbf{z}, \mathbf{z}_i) = 1$, in GEIDW it is the inverse of the correlation matrix that maintains this property. That is, the role of the inverse of the correlation matrix is to ensure the interpolation property while at the same time establishing another desired property, namely that the influence of the correction term diminishes as the distance of a considered input point to all training data points increases. This last property is not possessed by EIDW as discussed above.

It is easily checked that all properties of EIDW described in section 5.4 remain valid in the GEIDW setting.

7 Conclusion

In this paper we have developed a generalization of inverse distance weighting. In inverse distance weighting an approximation for an unknown value is obtained as a convex combination of known, correct output values in given input points. In the proposed generalization an approximation results as a correction to the value determined by another, given approximation method, where the correction is a convex combination of the components of an error vector. A further generalization is obtained by dropping the constraint of convexity, thereby allowing a more adequate trade-off between the prior approximation term and the correction term, by putting more weight to the prior approximation method as the considered input point moves away from all training data points, and vice versa. All fundamental properties of inverse distance weighting are essentially retained in this setting. The presented generalization of inverse distance weighting is equivalent to noise-free Gaussian process interpolation. The unique relationship between both methodologies is established by Riesz representation theorem for bounded, bilinear forms on the Cartesian product of a Hilbert space with itself.

References

[1] I. Andrianakis, P.G. Challenor, The effect of the nugget on Gaussian process emulators of computer models, Computational Statistics and Data Analysis 56 (2012) 4215-4228.

[2] F.J. Aguilar, JP. Mills, J. Delgado, M.A. Aguilar, J.G. Negreiros, J.L. Pérez, Modelling vertical error in LiDAR-derived digital elevation models, ISPRS Journal of Photogrammetry and Remote Sensing 65 (2010) 103-110.

[3] G. Allasia, Approximating potential integrals by cardinal basis interpolants on multivariate scattered data, Computer & Mathematics with Applications 43 (2002) 275-287.

[4] F. Chen, C. Liu, Estimation of the spatial rainfall distribution using inverse distance weighting (IDW) in the middle of Taiwan, Paddy and Water Environment 10 (2012) 209-222.

[5] N. Cressie, Geostatistics, The American Statistician 43 (1989) 197-202.

[6] L. de Mesnard, Pollution models and inverse distance weighting: some critical remarks, Computers & Geosciences 52 (2013) 459-469.

[7] W. Gossel, M. Falkenhagen, Line-geometry-based inverse distance weighted interpolation (L-IDW): geoscientific case studies, in: Proceedings of the 15th Annual Conference of the International Association for Mathematical Geosciences, 2015, pp. 333-337.

[8] S. Koziel, L. Leifsson, Surrogate-based modeling and optimization, first ed., Springer-Verlag, New York, 2013.

[9] L. Le Gratiet, Multi-fidelity Gaussian process regression for computer experiments, PhD thesis, 2013.

[10] A. O'Hagan, M.C. Kennedy, J.E. Oakley, Uncertainty analysis and other inference tools for complex computer codes, in: J.M. Bernardo, J.O. Berger, A.P. Dawid, A.F.M. Smith (Eds.), Bayesian Statistics 6, Oxford University Press, 1998, pp. 503-524.

[11] D.J. Leith, M. Heidl, J.V. Ringwood, Gaussian process prior models for electrical load forecasting, in: Proceedings of the International Conference on Probablistic Methods Applied to Power Systems, 2004.

[12] C.K.I. Williams, C.E. Rasmussen, Gaussian processes for regression, in: D.S. Touretzky, M.C. Mozer and M.E. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, MIT Press, 1996.

[13] R.M. Neal, Bayesian learning for neural networks, Springer, 1996.

[14] S. Särkkä, J. Hartikainen, Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression, in: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, 2012, pp. 993-1001.

[15] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: Proceedings of the 1968 ACM National Conference, 1968, pp. 517-524.

[16] M.L. Stein, Interpolation of spatial data: some theory for kriging, Springer, 1999.

[17] A. Verdin, B. Rajagopalan, W. Kleiber, A Bayesian kriging approach for blending satellite and ground precipitation observations, Water Resources Research 51 (2015) 908-921.

[18] G. Wahba, Improper priors, spline smoothing and the problem of guarding against model errors in regression, Journal of the Royal Statistical Society. Series B (Methodological) 40 (1978) 364-372.

[19] Y. Wang, B. Chaib-Draa, A KNN based kalman filter Gaussian process regression, in: roceedings of the Twenty-Third international joint conference on Artificial Intelligence, 2013, pp. 1771-1777.