



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
PROGRAMA DE MESTRADO E DOUTORADO EM CIÊNCIA DA
COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

DIEGO PARENTE PAIVA MESQUITA

MACHINE LEARNING FOR INCOMPLETE DATA

FORTALEZA

2017

DIEGO PARENTE PAIVA MESQUITA

MACHINE LEARNING FOR INCOMPLETE DATA

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Mestrado e Doutorado em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração:

Orientador: Prof. Dr. João P. P. Gomes

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M543m Mesquita, Diego Parente Paiva.

Machine Learning for Incomplete Data / Diego Parente Paiva Mesquita. – 2017.

55 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2017.

Orientação: Prof. Dr. João Paulo Pordeus Gomes.

1. Machine Learning. 2. Missing Data. I. Título.

CDD 005

DIEGO PARENTE PAIVA MESQUITA

MACHINE LEARNING FOR INCOMPLETE DATA

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Mestrado e Doutorado em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração:

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. João P. P. Gomes (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Antônio Macêdo
Universidade Federal do Ceará (UFC)

Prof. Dr. Aluizio F. R. Araújo
Universidade Federal de Pernambuco (UFPE)

Prof. Dr. André L. V. Coelho
Universidade de Fortaleza (Unifor)

To all of those who showed me love.

AGRADECIMENTOS

Como todas as coisas da vida, essa dissertação é uma consequência de vários acasos convenientes. Obrigado à todos que me propiciaram esses acasos. Eu extendo esse agradecimento à todos que passaram pela minha vida, mesmo que de maneira breve.

Obrigado aos meus pais, Moacir e Tânia, por sempre estarem comigo, me proverem de amor e me suportarem de todas as maneiras imagináveis.

Obrigado aos demais parentes, em especial ao meu amado avô Eulálio, que não terá a oportunidade de ler isso.

Obrigado aos meus queridos amigos - em especial Eiji Matsui, Luis Freitas e Rebeca Carneiro -, com os quais continuarei a dividir os bons momentos da vida.

Obrigado aos co-autores e amigos: João Paulo P. Gomes, Amauri H. Souza Jr. e Francesco Corona.

Obrigado à minha amada Rebeca Marcondes pelo cuidado, carinho e incentivo durante essa fase da minha vida.

Obrigado aos colegas, professores e demais funcionários do DC-UFC, que sempre fizeram que eu me sentisse confortável e bem-vindo.

Finalmente, obrigado aos meus gatos, Tupi e Tainá.

”First things first... second things never.”

(Who cares?)

RESUMO

Métodos baseados em funções de base (como as funções sigmoid e a q -Gaussian) e medidas de similaridade (como distâncias ou funções de kernel) são comuns em Aprendizado de Máquina e áreas correlatas. Comumente, no entanto, esses métodos não são equipados para utilizar dados incompletos de maneira orgânica. Isso pode ser visto como um impedimento, uma vez que dados parcialmente observados são comuns em vários domínios, como aplicações médicas e dados provenientes de sensores.

Nesta dissertação, propomos metodologias para estimar o valor do kernel Gaussiano, da distância Euclidiana, do kernel Epanechnikov e de funções de base arbitrárias na presença de vetores possivelmente parcialmente observados. Para obter tais estimativas, os vetores incompletos são tratados como variáveis aleatórias contínuas e, baseado nisso, tomamos o valor esperado da transformada de interesse.

Palavras-chave: Aprendizado de Máquina. Dados Incompletos. Kernel Gaussiano. Distância Euclidiana. Kernel Epanechnikov. Funções de base.

ABSTRACT

Methods based on basis functions (such as the sigmoid and q -Gaussian functions) and similarity measures (such as distances or kernel functions) are widely used in machine learning and related fields. These methods often take for granted that data is fully observed and are not equipped to handle incomplete data in an organic manner. This assumption is often flawed, as incomplete data is a fact in various domains such as medical diagnosis and sensor analytics. Therefore, one might find it useful to be able to estimate the value of these functions in the presence of partially observed data.

We propose methodologies to estimate the Gaussian Kernel, the Euclidean Distance, the Epanechnikov kernel and arbitrary basis functions in the presence of possibly incomplete feature vectors. To obtain such estimates, the incomplete feature vectors are treated as continuous random variables and, based on that, we take the expected value of the transforms of interest.

Keywords: Machine Learning. Missing Data. Gaussian Kernel. Euclidean Distance. Epanechnikov Kernel. Basis functions.

LIST OF TABLES

Table 2 – Overview of the experiments.	29
Table 3 – Kernel Estimates using different methods	30
Table 4 – Data sets description	31
Table 5 – Gaussian kernel estimation on real-world data: average RMSE	32
Table 6 – Overview of the experiments.	35
Table 7 – Euclidean distance estimates.	36
Table 8 – Euclidean distance estimation on real-world data: average RMSE	38
Table 9 – Overview of the experiments.	41
Table 10 – Epanechnikov kernel estimates.	42
Table 11 – Data sets description	42
Table 12 – Comparison between Expected Epanechnikov Kernel (EEK) and other methods.	44
Table 13 – Overview of the experiments.	49
Table 14 – Sigmoid function estimates.	50
Table 15 – q -Gaussian function estimates.	50
Table 16 – Data sets description	50
Table 17 – Comparison between SUnscented Transform (UT) and other methods to compute the sigmoid function on real-world data - RMSE values.	51
Table 18 – Comparison between SUT and other methods to compute the q -Gaussian function on real-world data - RMSE values.	52

LIST OF ACRONYMS

CMI	Conditional Mean Imputation
EEK	Expected Epanechnikov Kernel
EGK	Expected Gaussian Kernel
EM	Expectation Maximization
ESD	Expected Square Distance
GMM	Gaussian Mixture Model
ICkNNI	Incomplete-Case k -Nearest-Neighbors Imputation algorithm
MAR	Missing at Random
MC	Monte Carlo
MCAR	Missing Completely at Random
MNAR	Missing Not at Random
RBF	Radial Basis Function
RMSE	Root Mean Square Error
SLFNN	Single-Layer Feedforward Neural Network
UT	Unscented Transform

LIST OF SYMBOLS

μ	Mean vector
σ	Standard deviation
Σ	Covariance Matrix
\mathcal{X}	Set of feature vectors
X	Feature vector
\mathcal{N}	(Multivariate) Normal <i>p.d.f.</i>
\mathbb{E}	Expected value
Cov	Covariance
Var	Variance
Γ	Incomplete Gamma Function
f_{σ}	Sigmoid function
k	Kernel function
$\ \cdot\ ^2$	Euclidean norm
G	q -Gaussian function
e_q	q -exponential function
exp	Exponential
M	set of missing entries
O	set of observed entries
$O(\cdot)$	asymptotic notation

CONTENTS

1	INTRODUCTION	14
1.1	Objectives and chapters organization	15
1.2	Publications	15
1.2.1	<i>Directly related publications</i>	16
1.2.2	<i>Other contributions</i>	17
2	THEORETICAL BACKGROUND	19
2.1	Missing Data Terminology	19
2.2	Modelling data using GMMs	19
2.3	Expectation Maximization (EM) for Gaussian Mixture Model (GMM)s with Incomplete Data	21
2.4	Imputation Methods	23
2.4.1	<i>Conditional Mean Imputation</i>	23
2.4.2	<i>Incomplete-case K-NN Imputation</i>	24
3	EXPECTED GAUSSIAN KERNEL	25
3.1	Formulation	25
3.2	Experiments and Results	28
3.2.1	<i>EX1: Univariate Normal data with known parameters</i>	30
3.2.2	<i>EX2: Experiments on Real-World Data</i>	31
3.3	Conclusion	33
4	EXPECTED EUCLIDEAN DISTANCE	34
4.1	Formulation	34
4.2	Experiments and Results	35
4.2.1	<i>EX1: Univariate Normal data with known parameters</i>	36
4.2.2	<i>EX2: Experiments on Real-World Data</i>	36
4.3	Conclusion	39
5	EPANECHNIKOV KERNEL	40
5.1	Formulation	40
5.2	Experiments and Results	41
5.2.1	<i>EX1: Univariate Normal data with known parameters</i>	42
5.2.2	<i>EX2: Experiments on Real-World Data</i>	42

5.3	Conclusion	43
6	EXPECTED VALUE OF BASIS FUNCTIONS	45
6.1	Formulation	46
6.1.1	<i>Sigmoid Function</i>	47
6.1.2	<i>q-Gaussian Function</i>	48
6.2	Experiments and Results	48
6.2.1	<i>EX1: Univariate Normal data with known parameters</i>	49
6.2.2	<i>EX2: Experiments on Real-World Data</i>	50
6.3	Conclusion	52
7	CONCLUDING REMARKS	53
	REFERENCES	54

1 INTRODUCTION

Data completeness is a major assumption of most Machine Learning methods. In real world problems, however, several data instances may suffer from unobserved/missing attributes. This issue, referred to as missing/incomplete data problem, may happen due to a variety of reasons such as sensor problems, device malfunction and operator mistakes (EIROLA *et al.*, 2014). The simplest way to deal with missing data consists of removing the instances with missing attributes (listwise deletion) from the dataset. Even though this approach may work in some cases, discarding data samples usually leads to loss of important information to build a learning model (EIROLA *et al.*, 2013). Another widely used approach is to perform a pre-processing step of missing data imputation. After filling the missing entries, any conventional learning method can be used. Examples of such an approach can be found in (KANG, 2013), (LOBATO *et al.*, 2015), (ASTE *et al.*, 2015) and (GHEYAS; SMITH, 2010).

According to Acuña and Rodrigues in (ACUÑA; RODRIGUEZ, 2004), problems with more than 5% of missing samples may require sophisticated handling methods. In such situations, good results can be achieved by not considering the imputation as a separate step. Rather, it is possible to design a learning method that can handle incomplete data in its formulation. By doing so, the inherent uncertainty of the imputation process is taken into account and it has shown to be beneficial in many cases (SOVILJ *et al.*, 2016). On the other hand, direct imputation omits this uncertainty, which might be prejudicial depending on the context.

For example, let $X_i = (x_{i,1}, \dots, x_{i,D})^T$ and $X_j = (x_{j,1}, \dots, x_{j,D})^T$ be two (independent) possibly incomplete feature vectors. Suppose we are interested in estimating the squared Euclidean distance $\|X_i - X_j\|^2$ between these vectors, which is an important piece in many Machine Learning methods, such as Nearest Neighbors methods and k -means. Treating the missing entries as independent random variables, according to Eirola *et al.* (2013), the desired expected value can be expressed as:

$$\mathbb{E}[\|X_i - X_j\|^2] = \sum_{d=1}^D (\mathbb{E}[x_{i,d}] - \mathbb{E}[x_{j,d}])^2 + \text{Var}[x_{i,d}] + \text{Var}[x_{j,d}],$$

while imputing the missing entries of X_i and X_j with their expected value, would result

lead to:

$$\|\mathbb{E}[X_i] - \mathbb{E}[X_j]\|^2 = \sum_{d=1}^D (\mathbb{E}[x_{i,d}] - \mathbb{E}[x_{j,d}])^2,$$

which ignores the variances respective to the missing entries and, depending on their magnitude, might grossly diverge from $\mathbb{E}[\|X_i - X_j\|^2]$.

In this work, we provide tools to directly estimate transforms of incomplete feature vectors, in the a similar fashion to what (EIROLA *et al.*, 2013) proposed for the squared Euclidean distance.

1.1 Objectives and chapters organization

The general objective of this work is to provide tools for adapting Machine Learning methods which incorporate the uncertainty arising from the imputation process, similarly to what was presented in (EIROLA *et al.*, 2013) and Eirola *et al.* (2014) for the squared Euclidean distance. We present these in the form of methodologies to estimate transforms of the values which would be otherwise imputed. As specific objectives, we address the problems of estimating:

1. the Gaussian Kernel between two possibly incomplete feature vectors;
2. the Euclidean Distance between two possibly incomplete feature vectors;
3. the Epanechnikov Kernel between two possibly incomplete feature vectors;
4. the value of basis functions applied to a possibly incomplete feature vector.

A theoretical background covering basic concepts and commonly used imputation methods is provided in chapter 2. Solutions to the aforementioned specific objectives are presented, in order, in chapters 3, 4 and 5. In chapter 7, we provide final reflections on the content of the presented work and discuss directions for future works.

1.2 Publications

During the span in which the work presented here was in development, a number of articles have been published by the author. This includes articles which directly relate to the thesis topic, as well as articles which are products of cooperation with colleagues and faculty from diverse research areas. The former ones are listed in Subsection 1.2.1, while the remaining ones are outlined in Subsection 1.2.2

1.2.1 *Directly related publications*

1. Diego P. P. Mesquita, João P. P. Gomes, Amauri H. Souza Junior **Epanechnikov Kernel for Incomplete Data**
Electronics Letters, 2017.
2. Diego P. P. Mesquita, João P. P. Gomes, Amauri H. Souza Junior, Juvêncio S. Nobre **Euclidean Distance Estimation in Incomplete Datasets**
Neurocomputing, 2017.
3. Marcelo B. Veras, Diego P. P. Mesquita, João P. P. Gomes, Amauri H. Souza Junior, Guilherme A. Barreto
Forward Stagewise Regression on Incomplete datasets
International Work-conference on Artificial Neural Networks, 2017.
4. Diego P. P. Mesquita, João P. P. Gomes, Leonardo R. Rodrigues
K-means for Datasets with Missing Attributes: Building Soft Constraints with Observed and Imputed Values
European Symposium on Artificial Neural Networks, 2016.
5. Diego P. P. Mesquita, João P. P. Gomes, Leonardo R. Rodrigues
Extreme Learning Machines for Datasets with Missing Values Using the Unscented Transform
Brazilian Conference on Intelligent Systems, 2016 .
6. Diego P. P. Mesquita, João P. P. Gomes
Radial Basis Function Neural Networks for datasets with missing values
International Conference on Intelligent Systems Design and Applications, 2016 .
7. Diego P. P. Mesquita, João P. P. Gomes, Amauri H. Souza Junior **A Minimal Learning Machine for datasets with missing values**
International Conference on Neural Information Processing, 2015.
8. Diego P. P. Mesquita, João P. P. Gomes, Francesco Corona, Amauri H. Souza Junior, Juvêncio S. Nobre
Gaussian Kernels for Incomplete Data
Information Sciences (second round of reviews).

1.2.2 Other contributions

1. Diego P. P. Mesquita, João P. P. Gomes, Leonardo R. Rodrigues, Saulo A. F. Oliveira and Roberto K. H. Galvão
Building Selective Ensembles of Randomization Based Neural Networks with the Successive Projections Algorithm
Applied Soft Computing, 2017.
2. Diego P. P. Mesquita, João P. P. Gomes, Amauri H. Souza Junior
Ensemble of Efficient Minimal Learning Machines for Classification and Regression
Neural Processing Letters, 2017.
3. Diego P. P. Mesquita, Lincoln S. Rocha, João P. P. Gomes, Ajalmar R. Rocha Neto
Classification with Reject Option for Software Defect Prediction
Applied Soft Computing, 2016.
4. Diego P. P. Mesquita, João P. P. Gomes, Leonardo R. Rodrigues, Roberto K. H. Galvão
Pruning Extreme Learning Machines Using the Successive Projections Algorithm
IEEE Latin America Transactions, 2015.
5. João P. P. Gomes, Diego P. P. Mesquita, Ananda L. Freire, Amauri H. Souza Junior, Tommi Karkkainen
A Robust Minimal Learning Machine based on the M-Estimator
European Symposium on Artificial Neural Networks, 2017.
6. Diego P. P. Mesquita, Antônio N. Araújo Neto, José F. Queiroz Neto, João P. P. Gomes, Leonardo R. Rodrigues
Using Robust Extreme Learning Machines to Predict Cotton Yarn Strength and Hairiness
European Symposium on Artificial Neural Networks, 2016.
7. Filipe F. R. Damasceno, Diego P. P. Mesquita, Marcelo B. Veras, João P. P. Gomes, Carlos E. F. de Brito
Shrinkage k-means: A clustering algorithm based on the James-Stein Estimator
Brazilian Conference on Intelligent Systems, 2016 .

8. Wesley L. Caldas, Michelle G. Cacaís, João P. P. Gomes, Diego P. P. Mesquita
Co-MLM: a SSL algorithm based on the Minimal Learning Machine
Brazilian Conference on Intelligent Systems, 2016 .
9. Diego P. P. Mesquita, João P. P. Gomes, Amauri H. Souza Junior, Guilherme A. Barreto
Ensemble of Minimal Learning Machine for Pattern Classification
International Work-conference on Artificial Neural Networks, 2015.

2 THEORETICAL BACKGROUND

In this chapter, we introduce some basic concepts which might be useful to understanding the developments presented in the next chapters. In Section 2.1, we introduce basic missing data concepts regarding the processes which causes entries to be missing. In Section 2.2, we introduce the Gaussian Mixture Model (GMM) and address how to use it to obtain statistics of missing components in a vector. In Section 2.3, we introduce Expectation Maximization (EM) as a procedure to estimate the parameters of a GMM from incomplete datasets. Finally, in Section 2.4, we give an overview of two popular imputation strategies that will be used for comparison.

2.1 Missing Data Terminology

Consider a D -dimensional random vector X with observed entries indexed as X_O and missing ones as X_M . Let $I \in \{0, 1\}^D$ be an indicator vector such that the I_n equals one if and only if X_n is observed, i.e., $n \in O$. We say data is Missing Completely at Random (MCAR) if the probability an entry is missing in X is independent of values of both the observed and missing entries of X , which can be expressed as:

$$P(I|X_O, X_M) = P(I).$$

Differently from MCAR, data is said to be Missing at Random (MAR) when the probability that a component X_n of X is missing is independent of its true (unknown) value, but might depend of X_O . This can be expressed as:

$$P(I|X_O, X_M) = P(I|X_O)$$

When the probability that an entry is missing is intrinsically related to its value, it is said that data is Missing Not at Random (MNAR). In this work, we consider data is MAR. A more comprehensive account of missing data mechanisms can be found in Molenberghs *et al.* (2014).

2.2 Modelling data using GMMs

To provide a flexible representation for the distribution from which the feature vectors in a dataset $\mathcal{X} = \{X_n\}_{n=1}^N \subset \mathbb{R}^D$ were drawn, we assume it can be modelled as a

linear superposition of C D -dimensional Gaussian densities, each with its own mean $\mu^{(c)}$ and covariance matrix $\Sigma^{(c)}$, with $c = 1, \dots, C$, i.e., a Gaussian Mixture Model (GMM). Given an arbitrary $X_n \in \mathbb{R}^D$, the probability density function of a GMM (HUNT; JORGENSEN, 2003) with the aforementioned parameters takes the form:

$$p(X_n) = \sum_{c=1}^C w^{(c)} \mathcal{N}(X_n | \mu^{(c)}, \Sigma^{(c)}), \quad (2.1)$$

where $\{w^{(c)}\}_{c=1}^C$ is a set of non-negative scalars that satisfy the convexity constraint $\sum_{c=1}^C w^{(c)} = 1$. The GMM model is a flexible and powerful modeling tool capable to model a wide class of continuous distributions, provided a sufficient number of Gaussian components.

It is instrumental for the developments in this work to be able to obtain estimates of the non-central moments of the missing entries of a feature vector. It is known that, for a single Gaussian ($C = 1$) these moments can be expressed as functions of its mean vector and covariance matrix. In turn, the mean and covariance of the missing entries can be computed by conditioning the Gaussian on the observed entries of the vector. In a GMM, the moments are given by a weighted sum of the moments of the Gaussian components.

For instance, consider an arbitrary vector X_n , with missing component values $X_{n,M}$ and observed component values $X_{n,O}$, where M and O denote the sets of indexes of missing and observed component values, respectively. Then, the parameters $\mu^{(c)}$ and $\Sigma^{(c)}$ of the c -th component of the GMM can be partitioned into blocks as follows:

$$\mu^{(c)} = \begin{bmatrix} \mu_O^{(c)} \\ \mu_M^{(c)} \end{bmatrix}, \quad \Sigma^{(c)} = \begin{bmatrix} \Sigma_{OO}^{(c)} & \Sigma_{OM}^{(c)} \\ \Sigma_{MO}^{(c)} & \Sigma_{MM}^{(c)} \end{bmatrix}. \quad (2.2)$$

Then, the mean vector $\tilde{\mu}_n^{(c)} = \mathbb{E}^{(c)}[X_{n,M} | X_{n,O}]$ and covariance matrix $\tilde{\Sigma}_n^{(c)} = \text{Var}^{(c)}[X_{n,M} | X_{n,O}]$ of the c -th Gaussian in the GMM, conditioned on $X_{n,O}$, are given by

$$\tilde{\mu}_n^{(c)} = \mu_M^{(c)} + \Sigma_{MO}^{(c)} (\Sigma_{OO}^{(c)})^{-1} (X_{n,O} - \mu_O^{(c)}), \quad (2.3)$$

$$\tilde{\Sigma}_n^{(c)} = \Sigma_{MM}^{(c)} - \Sigma_{MO}^{(c)} (\Sigma_{OO}^{(c)})^{-1} \Sigma_{OM}^{(c)}, \quad (2.4)$$

and the first four non-central moments of a particular component $x_{n,d}$ of $X_{n,M}$ are given

by

$$\mathbb{E}[x_{n,d}] = \sum_{c=1}^C w^{(c)} \tilde{\mu}_{n,d}^{(c)}, \quad (2.5)$$

$$\mathbb{E}[x_{n,d}^2] = \sum_{c=1}^C w^{(c)} \left([\tilde{\mu}_{n,d}^{(c)}]^2 + \tilde{\Sigma}_{n,d}^{(c)} \right), \quad (2.6)$$

$$\mathbb{E}[x_{n,d}^3] = \sum_{c=1}^C w^{(c)} \left([\tilde{\mu}_{n,d}^{(c)}]^3 + 3\tilde{\mu}_{n,d}^{(c)} \tilde{\Sigma}_{n,d}^{(c)} \right), \quad (2.7)$$

$$\mathbb{E}[x_{n,d}^4] = \sum_{c=1}^C w^{(c)} \left([\tilde{\mu}_{n,d}^{(c)}]^4 + 6[\tilde{\mu}_{n,d}^{(c)}]^2 \tilde{\Sigma}_{n,d}^{(c)} + 3[\tilde{\Sigma}_{n,d}^{(c)}]^2 \right), \quad (2.8)$$

in which $\tilde{\mu}_{n,d}^{(c)}$ denotes the d -th element of vector $\tilde{\mu}_n^{(c)}$ and $\tilde{\Sigma}_{n,d}^{(p)}$ denotes the d -th element along the main diagonal of matrix $\tilde{\Sigma}_n^{(p)}$.

2.3 EM for GMMs with Incomplete Data

Expectation-maximization algorithms are efficient approaches for finding a maximum likelihood solution for models with latent variables as a Gaussian mixture model, without relying on iterative numerical optimization techniques. Given a likelihood function of some parameters, EM algorithms consist of two steps, an expectation and a maximization step: In the first step, the expected value of some latent variables is taken; in the latter, the most likely estimates for the parameters are computed. The two steps are repeated until convergence of either the parameters or the likelihood. We briefly overview the EM algorithm for GMMs with complete data and its extension for incomplete data (HUNT; JORGENSEN, 2003).

Consider a data set $\mathcal{X} = \{X_n\}_{n=1}^N$ comprising N samples and the Gaussian mixture distribution $p(X_n) = \sum_{c=1}^C w^{(c)} \mathcal{N}(X_n | \mu^{(c)}, \Sigma^{(c)})$ consisting of C densities $\mathcal{N}(X_n | \mu^{(c)}, \Sigma^{(c)})$. Let $\Theta = \{w^{(c)}, \mu^{(c)}, \Sigma^{(c)}\}_{c=1}^C$ with $w^{(c)}$ as mixing coefficient of the c -th Gaussian and $\mu^{(c)}$ and $\Sigma^{(c)}$ its mean vector and covariance matrix. We want to maximize the likelihood $\mathcal{L}_{\mathcal{X}}(\Theta)$ of the parameters Θ

$$\mathcal{L}_{\mathcal{X}}(\Theta) = \prod_{n=1}^N \left(\sum_{c=1}^C w^{(c)} \mathcal{N}(X_n | \mu^{(c)}, \Sigma^{(c)}) \right). \quad (2.9)$$

After initializing the means, the covariances and the mixing coefficients, and calculating the initial value of the likelihood, an expectation-maximization algorithm sequentially repeats the expectation and the maximization steps:

1. The expectation (E) step computes the expected memberships $t_{n,c}$ of each sample n

$$t_{n,c} = \frac{w^{(c)} \mathcal{N}(X_n | \mu^{(c)}, \Sigma^{(c)})}{\sum_l w^{(l)} \mathcal{N}(X_n | \mu^{(l)}, \Sigma^{(l)})}, \quad (2.10)$$

with respect to each Gaussian $c = 1, \dots, C$.

2. The maximization (M) step consists of computing the most likely estimates for the parameters in Θ

$$\mu^{(c)} = \frac{1}{N_c} \sum_{n=1}^N t_{n,c} X_n, \quad (2.11)$$

$$\Sigma^{(c)} = \frac{1}{N_c} \sum_{n=1}^N t_{n,c} (X_n - \mu^{(c)})(X_n - \mu^{(c)})^T, \quad (2.12)$$

$$w^{(c)} = \frac{N_c}{N}. \quad (2.13)$$

with $N_c = \sum_{n=1}^N t_{n,c}$ and for $c = 1, \dots, C$.

Although the formulation is a well-established approach to fit Gaussian mixture models to complete data, a number of modifications are required to extend it to the case of incomplete data. The idea consists in treating missing values as latent variables to be estimated in the expectation step. The likelihood given the observed values takes the form

$$\mathcal{L}_{\mathcal{X}}(\Theta) = \prod_{n=1}^N \left(\sum_{c=1}^C w^{(c)} \mathcal{N}(X_{n,O} | \mu_O^{(c)}, \Sigma_{OO}^{(c)}) \right), \quad (2.14)$$

where $X_{n,O}$ denotes the observed entries of X_n and $\{\mu_O^{(c)}, \Sigma_{OO}^{(p)}\}_{c=1}^C$ are the parameters of the c -th Gaussian when marginalizing on the observed entries of X_n .

To make use of the marginal probability on the observed entries, the E-step in Eq. (2.10) is modified to yield

$$t_{n,c} = \frac{w^{(c)} \mathcal{N}(X_{n,O} | \mu_O^{(c)}, \Sigma_{OO}^{(c)})}{\sum_l w^{(l)} \mathcal{N}(X_{n,O} | \mu_O^{(l)}, \Sigma_{OO}^{(l)})}. \quad (2.15)$$

The E-step is further augmented with the following equations that compute the parameters $\{\tilde{\mu}_n^{(c)}, \tilde{\Sigma}_n^{(c)}\}_{c=1}^C$ of the distribution of the missing entries $X_{n,M}$ conditioned on $X_{n,O}$, i.e $\tilde{\mu}_n^{(c)} = \mathbb{E}^{(c)}[X_{n,M} | X_{n,O}]$ and $\tilde{\Sigma}_n^{(c)} = \text{Var}^{(c)}[X_{n,M} | X_{n,O}]$. For each Gaussian:

$$\tilde{\mu}_n^{(c)} = \mu_{n,M}^{(c)} + \Sigma_{MO}^{(c)} (\Sigma_{OO}^{(c)})^{-1} (X_{n,O} - \mu_{n,O}^{(c)}),$$

$$\tilde{\Sigma}_n^{(c)} = \Sigma_{MM}^{(c)} - \Sigma_{MO}^{(c)} (\Sigma_{OO}^{(c)})^{-1} \Sigma_{OM}^{(c)}.$$

As for the M-step, Eq. (2.13) that computes the weights of the mixture remains functionally unaltered, while Eqs. (2.11) and (2.12) are modified to yield

$$\begin{aligned}\mu^{(c)} &= \frac{1}{N_c} \sum_{n=1}^N t_{n,c} \tilde{X}_n^{(c)}, \\ \Sigma^{(c)} &= \frac{1}{N_c} \sum_{n=1}^N t_{n,c} (\tilde{X}_n^{(c)} - \mu^{(c)})(\tilde{X}_n^{(c)} - \mu^{(c)})^T + \sum_{n=1}^N t_{n,c} \Sigma_n^{(c)},\end{aligned}$$

where:

$$\tilde{X}_n^{(c)} = \begin{bmatrix} X_{n,O} \\ \tilde{\mu}_n^{(c)} \end{bmatrix}, \quad \Sigma_n^{(c)} = \begin{bmatrix} \mathbf{0}_{OO} & \mathbf{0}_{OM} \\ \mathbf{0}_{MO} & \tilde{\Sigma}_n^{(c)} \end{bmatrix}. \quad (2.16)$$

In other words, $\tilde{X}_n^{(c)}$ denotes the vector X_n imputed with $\tilde{\mu}_n^{(c)}$ on its missing entries and $\Sigma_n^{(c)}$ is the conditional covariance matrix $\tilde{\Sigma}_n^{(c)}$ padded with zeros.

2.4 Imputation Methods

2.4.1 Conditional Mean Imputation

Conditional Mean Imputation (CMI) consists in estimating a probability distribution for the missing entries in a vector and using its mean value to fill the entries. The usual approach to obtain such distribution is to first obtain a model for the distribution from which the feature vectors of the dataset were drawn and then condition on the observed values of each incomplete vector.

Let p denote the probability density function obtained from a dataset \mathcal{X} . Given an incomplete vector $X \in \mathcal{X}$, CMI consists in filling each missing entry $x_{i,k}$ of X with:

$$\int_{-\infty}^{\infty} \phi p(\phi | x_{i,O}) d\phi \quad (2.17)$$

where $p(\cdot | x_{i,O})$ denotes the p.d.f. p conditioned on the observed entries of X .

For the case in which p is the p.d.f. of a Gaussian distribution, so is the conditional $p(\cdot | x_{i,O})$, which can be obtained as described in Section 2.2. If the parameters of $p(\cdot | x_{i,O})$ are μ (mean vector) and Σ (covariance matrix), Eq. 2.17 resumes to μ .

Similarly, if a GMM is used to model the data, Eq. 2.17 becomes a weighted sum of the components of the conditioned components of the GMM. This process was also addressed in Section 2.2.

2.4.2 Incomplete-case K -NN Imputation

Incomplete-case k Nearest Neighbors imputation (ICkNNI) is a non-parametric distance-based imputation method recently proposed by Hulse e Khoshgoftaar (2014). The idea behind ICkNNI is to use the incomplete Euclidean distance in order to select the k nearest fully observed neighbors for each incomplete feature vector, then filling the missing entries with the average of the selected neighbors. In this Section , we provide a review of ICkNNI.

Consider a set $\mathcal{X} = \{X_n\}_{n=1}^N$ comprising both complete and incomplete feature vectors. Let $x_{i,k}$ denote the k -th component of X_i and O_i be the set of indices of the observed entries in X_i . Furthermore, let

$$\mathfrak{N}_{i,j} = \{X_n \in \mathcal{X} | O_i \cup \{j\} \subseteq O_n\} \quad (2.18)$$

be the set of points in \mathcal{X} which count on all the observed features of X_i plus feature $j \in M_i$.

The incomplete euclidean distance d can be defined as:

$$d(X_i, X_j) = \begin{cases} \sqrt{\sum_{k \in O_i} (x_{i,k} - x_{j,k})^2}, & \text{if } O_i \subseteq O_j; \\ \infty, & \text{otherwise.} \end{cases} \quad (2.19)$$

In Incomplete-Case k -Nearest-Neighbors Imputation algorithm (ICkNNI), for each incomplete feature vector $X_i \in \mathcal{X}$, and $l \in O_i$, we construct the set $\mathfrak{C}_{i,j}$ consisting of the k nearest neighbors of X_i in the set $\mathfrak{N}_{i,j}$ according to $d(X_i, \cdot)$. Then we fill the missing component $x_{i,j}$ with:

$$\sum_{X_j \in \mathfrak{C}_{i,j}} \frac{x_{j,l}}{k}. \quad (2.20)$$

3 EXPECTED GAUSSIAN KERNEL

The Gaussian Kernel, often referred to as the Radial Basis Function (Radial Basis Function (RBF)) kernel is one of the most common kernels in Machine Learning, finding application in various methods, such as Support Vector Machines, Gaussian Processes and RBF networks.

Given two D -dimensional vectors $X_i = (x_{i,1}, \dots, x_{i,D})^T$ and $X_j = (x_{j,1}, \dots, x_{j,D})^T$, the Gaussian kernel is given by

$$k(X_i, X_j) \triangleq \exp \left\{ - \frac{\|X_i - X_j\|^2}{2\sigma^2} \right\}, \quad (3.1)$$

where $\sigma^2 > 0$ is the scale hyper-parameter and $\|X_i - X_j\|^2 = \sum_{d=1}^D (x_{i,d} - x_{j,d})^2$, i.e., $\|\cdot\|$ denotes the L2 norm.

In this Chapter, we present a methodology to estimate $k(X_i, X_j)$ when vectors $X_i, X_j \in \mathcal{X}$ count on one or more missing components.

3.1 Formulation

Let X_i and X_j be two possibly partially observed feature vectors drawn from a same distribution and let $z = \|X_i - X_j\|^2$. Note z is a random variable as it is a transform of the missing entries of X_i and X_j . In turn, so is $k(X_i, X_j)$, since it can be written as:

$$k(X_i, X_j) = \exp \left\{ - \frac{z}{2\sigma^2} \right\}. \quad (3.2)$$

Thus, estimating $k(X_i, X_j)$ comes down to computing:

$$\mathbb{E}[k(X_i, X_j)] = \mathbb{E} \left[\exp \left\{ - \frac{z}{2\sigma^2} \right\} \right] = \int_{-\infty}^{\infty} \exp \left\{ - \frac{z}{2\sigma^2} \right\} p_z(z) dz \quad (3.3)$$

where p_z denotes the probability density function of z . Furthermore, as z is non-negative:

$$\mathbb{E}[k(X_i, X_j)] = \int_0^{\infty} \exp \left\{ - \frac{z}{2\sigma^2} \right\} p_z(z) dz = \int_{(0, \infty)} \exp \left\{ - \frac{z}{2\sigma^2} \right\} p_z(z) dz. \quad (3.4)$$

Recall the definition of the moment-generating function M_z of a random variable z with p.d.f. p_z whose support is $(0, \infty)$:

$$M_z(t) = \int_{(0, \infty)} e^{tz} p_z(z) dz, \quad (3.5)$$

hence:

$$\mathbb{E}[k(X_i, X_j)] = M_z \left(-\frac{1}{2\sigma^2} \right), \quad (3.6)$$

i.e., computing $\mathbb{E}[k(X_i, X_j)]$ boils down to evaluating $M_z(\cdot)$ at $t = -1/(2\sigma^2)$. For such, we need to choose a distribution for z .

For $d = 1 \dots D$, let $\phi_d^2 = (x_{i,d} - x_{j,d})^2$. Consequently, since the differences ϕ_d can¹ be treated as random variables, $z = \sum_{d=1}^D \phi_d^2$ is a sum of squared random variables. According to Roberts e Geisser (1966), the distribution of a squared random variable ϕ_d^2 is said to be Gamma - with parameters α_d and β_d - if:

1. the distribution p_d of ϕ_d can be written as:

$$p_d(\phi_d) = h(\phi_d) |\phi_d|^{2\alpha_d-1} \exp(-\beta_d \phi_d^2);$$

2. There exists a constant ζ such that:

$$\forall \phi_d : h(\phi_d) + h(-\phi_d) = \zeta.$$

A variety of different distributions conform to the above conditions. Some examples are the Gaussian distribution, the Skew Normal (AZZALINI, 1985) (which generalizes the Gaussian distribution and allows for non-zero asymmetry), Kotz-type distributions (that are bi-modal and may have light tails) as well as other heavy- and light-tailed distributions obtained similarly to the Skew Normal (ROBERTS; GEISSER, 1966; JOHNSON N.; BALAKRISHNAN, 1995; GENTON, 2004).

It is then reasonable to model z as a sum of Gamma-distributed random variables. Furthermore, Covo e Elalouf (2014) showed the sum of independent Gamma distributed random variables can be well approximated by a Gamma distribution under mild conditions. This independence assumption is satisfied if we assume the missing components of a vector are independent among themselves. This premise can be found in (EIROLA *et al.*, 2013) and (EIROLA *et al.*, 2014), and we take it as well. Thus, we assume z follows a Gamma distribution with shape and inverse scale parameters, respectively, $\alpha, \beta > 0$. Hence, $p_z(\cdot)$ is given by:

$$p(z|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} z^{\alpha-1} \exp(-\beta z), \quad (3.7)$$

¹ In case both $x_{i,d}$ and $x_{j,d}$ are known, we could attribute a small variance to ϕ_d .

where $\Gamma(\alpha) = \int_0^{+\infty} u^{\alpha-1} \exp(-\alpha) du$ is the gamma function.

In particular, for the Gamma distribution, $M_z(\cdot)$ has a closed-form solution:

$$M_z(t) = \left(1 - \frac{t}{\beta}\right)^{-\alpha} \quad \forall t < \beta, \quad (3.8)$$

and Eq. (3.6) turns into:

$$\mathbb{E}[k(X_i, X_j)] = M_z\left(-\frac{1}{2\sigma^2}\right) = \left(\frac{2\beta\sigma^2}{2\beta\sigma^2 + 1}\right)^\alpha. \quad (3.9)$$

To estimate $k(X_i, X_j)$, we still have to estimate the parameters α and β of the Gamma distribution. These can be obtained via method-of-moments from the mean $\mathbb{E}[z]$ and the variance $\text{Var}[z]$ of the squared distance z :

$$\alpha = \frac{\mathbb{E}[z]^2}{\text{Var}[z]}, \quad \beta = \frac{\mathbb{E}[z]}{\text{Var}[z]}. \quad (3.10)$$

The problem of estimating $\mathbb{E}[z]$ can be approached using the results from (EIROLA *et al.*, 2013) and (EIROLA *et al.*, 2014), in which expected squared distances are expressed in the form

$$\begin{aligned} \mathbb{E}[z] = & \sum_{d \notin M_i \cup M_j} (x_{i,d} - x_{j,d})^2 + \sum_{d \in M_j \setminus M_i} \mathbb{E}[(x_{i,d} - x_{j,d})^2] \\ & + \sum_{d \in M_i \setminus M_j} \mathbb{E}[(x_{i,d} - x_{j,d})^2] + \sum_{d \in M_i \cap M_j} \mathbb{E}[(x_{i,d} - x_{j,d})^2], \end{aligned} \quad (3.11)$$

with $M_i, M_j \subseteq \{1, \dots, D\}$ denoting the sets of indexes of the missing components of X_i and X_j , respectively. Since all of the terms in eq. (3.11) can be expanded to yield:

$$\begin{aligned} \mathbb{E}[(x_{i,d} - x_{j,d})^2] &= \mathbb{E}[x_{i,d}^2] + \mathbb{E}[x_{j,d}^2] - 2\mathbb{E}[x_{i,d}]\mathbb{E}[x_{j,d}] \\ &= \begin{cases} \mathbb{E}[x_{i,d}^2] - \mathbb{E}[x_{i,d}]^2 + \mathbb{E}[x_{j,d}^2] - \mathbb{E}[x_{j,d}]^2 \\ + \mathbb{E}[x_{i,d}]^2 + \mathbb{E}[x_{j,d}]^2 - 2\mathbb{E}[x_{i,d}]\mathbb{E}[x_{j,d}] \end{cases} \\ &= (\mathbb{E}[x_{i,d}] - \mathbb{E}[x_{j,d}])^2 + \text{Var}[x_{i,d}] + \text{Var}[x_{j,d}], \end{aligned}$$

the expected squared distance z can be compactly written as

$$\mathbb{E}[z] = \sum_{d=1}^D (\mathbb{E}[x_{i,d}] - \mathbb{E}[x_{j,d}])^2 + \text{Var}[x_{i,d}] + \text{Var}[x_{j,d}]. \quad (3.12)$$

Although not derived in (EIROLA *et al.*, 2013) and (EIROLA *et al.*, 2014), analogous reasoning can be used to express the variance of the squared distances $\text{Var}[z]$:

$$\begin{aligned}\text{Var}[z] &= \text{Var}\left[\sum_{d=1}^D (x_{i,d} - x_{j,d})^2\right] \\ &= \sum_{d=1}^D \text{Var}[(x_{i,d} - x_{j,d})^2] + \sum_{d=1}^D \sum_{l=d+1}^D \text{Cov}[(x_{i,d} - x_{j,d})^2, (x_{i,l} - x_{j,l})^2].\end{aligned}\tag{3.13}$$

Using independence assumptions stated before, Eq. (3.13) reduces to:

$$\begin{aligned}\text{Var}[z] &= \sum_{d=1}^D \text{Var}[(x_{i,d} - x_{j,d})^2] \\ &= \sum_{d=1}^D \mathbb{E}[(x_{i,d} - x_{j,d})^4] - \mathbb{E}[(x_{i,d} - x_{j,d})^2]^2 \\ &= \begin{cases} \sum_{d=1}^D \mathbb{E}[x_{i,d}^4 + x_{j,d}^4 - 4x_{i,d}^3 x_{j,d} - 4x_{i,d} x_{j,d}^3 + 6x_{i,d}^2 x_{j,d}^2] \\ - \sum_{d=1}^D \mathbb{E}[(x_{i,d} - x_{j,d})^2]^2. \end{cases}\end{aligned}\tag{3.14}$$

Together, eqs. (3.12) and (3.14) show how the expectation and the variance of squared distances can be expressed only in terms of non-central moments of X_i and X_j . Such moments can be estimated by imposing a distribution from which X_i and X_j are drawn and estimating the parameters of such distribution. Any model-estimation method capable of generating probability distributions for each missing variable can be used for the task.

3.2 Experiments and Results

We perform two different experiments to validate our approach, the Expected Gaussian Kernel (EGK). In the first, we study how the uncertainty on the estimation of the missing values affects the quality of the kernel estimate. In second, we evaluate EGK on real-world data. Table 2 summarizes the details of these experiments.

In the first experiment, EGK is compared against Conditional Mean Imputation (CMI) (HUNT; JORGENSEN, 2003) and Expected Square Distance (ESD) (EIROLA *et al.*, 2013). It is interesting to notice that these methods differ from EGK mainly in the level in which the estimation problem is cast. In CMI, the values of the missing entries of X_i and X_j are estimated (and later used to compute $k(X_i, X_j)$). The ESD approach

Table 2 – Overview of the experiments.

Objective	Setup
EX1 Assess the quality of the kernel estimation as a function of the uncertainty on the estimation of the missing values.	X_i and X_j are drawn from an univariate Normal distribution with known mean and variance, but X_i missing.
EX2 Validate the method on Real-world data.	Different datasets from the UCI repository were employed.

consists in estimating $\|X_i - X_j\|^2$ and plugging it into the kernel expression. On the other hand, EGK directly estimates the transform of interest. This conceptual difference between the approaches is condensed in Eqs. (3.15) to (3.17).

$$\hat{k}_{EGK}(X_i, X_j) = \mathbb{E} \left[\exp \left\{ - \frac{\|X_i - X_j\|^2}{2\sigma^2} \right\} \right], \quad (3.15)$$

$$\hat{k}_{ESD}(X_i, X_j) = \exp \left\{ - \frac{\mathbb{E}[\|X_i - X_j\|^2]}{2\sigma^2} \right\}, \quad (3.16)$$

$$\hat{k}_{CMI}(X_i, X_j) = \exp \left\{ - \frac{\|\mathbb{E}[X_i] - \mathbb{E}[X_j]\|^2}{2\sigma^2} \right\}. \quad (3.17)$$

It has been pointed out by Eirola *et al.* (2013) that estimating the missing entries before taking the squared euclidean distance tends to underestimate the expected value of this transform. As a consequence - see Eq. (3.12) -, we have

$$\hat{k}_{ESD}(X_i, X_j) \leq \hat{k}_{CMI}(X_i, X_j). \quad (3.18)$$

Assuming z is Gamma-distributed, a similar statement can be made via direct application of Jensen's inequality. Recall Jensen's inequality states that:

$$g(\mathbb{E}[\phi]) \leq \mathbb{E}[g(\phi)] \quad (3.19)$$

for any integrable real-valued random variable ϕ and convex function $g(\cdot)$. Applying it directly to Eq. (3.15), one can obtain

$$\hat{k}_{ESD}(X_i, X_j) \leq \hat{k}_{EGK}(X_i, X_j). \quad (3.20)$$

3.2.1 EX1: Univariate Normal data with known parameters

For this experiment, we fix $X_j = 3$, assume $X_i \sim \mathcal{N}(2, \sigma_n^2)$ and estimate $k(X_i, X_j)$. Since the distribution of X_i is known, there is no need to estimate a model for the data as the true distribution $\mathcal{N}(2, \sigma_n^2)$ is given. We set the kernel hyper-parameter $\sigma^2 = 1$.

To obtain a benchmark, we compute a Monte Carlo (Monte Carlo (MC)) estimate of $k(X_i, X_j)$ by performing 10^8 draws of X_i from $\mathcal{N}(2, \sigma_n^2)$, taking the kernel value for each of these draws and then averaging the computed kernel values. Using this procedure, we aim to accurately approximate the expected value the kernel. Based on that, a method is as good as its estimates are similar to the ones obtained by via Monte Carlo. Table 3 compiles the results of the experiments for different values of σ_n^2 .

Table 3 – Kernel Estimates using different methods

σ_n^2	MC	CMI	ESD	EGK
10^{-2}	0.6065	0.6065	0.6035	0.6065
10^{-1}	0.6052	0.6065	0.5769	0.6045
10^0	0.5507	0.6065	0.3679	0.5429
10^1	0.2881	0.6065	0.0041	0.2868
10^2	0.0990	0.6065	1.1698e-22	0.0990

Note that CMI computes the same approximation regardless of the value of σ_n^2 . This is expected, since the the expected value of X_i depends only of μ . While both CMI and ESD quickly deteriorate as σ_n^2 increases, EGK maintains a steady performance, approximating the Monte Carlo estimate more accurately. It is also interesting to notice that the results above ratifies Eqs. (3.18) and (3.20).





3.2.2 EX2: Experiments on Real-World Data

In this second experiment, we evaluate the performance of EGK in real world data sets for different amounts of missing data. All data used for this experiments is available at the UCI repository of Machine Learning data sets (LICHMAN, 2013). Details concerning the number and size of the feature vectors in each of data set can be found in Table 4.

Table 4 – Data sets description

Dataset	Size	Features
MPG	392	7
FOREST-FIRE (FIRE)	517	4
COLUMN (COL)	310	6
HABERMAN (HAB)	306	3
DIABETES (PID)	768	8
IRIS	150	4
CONCRETE COMPRESSION (COMP)	1030	80
CONCRETE SLUMP (SLUMP)	103	8

Ten similar rounds of experiment were carried. In each of these, the percentage of instances with missing vectors was iteratively increased from 10% to 50% (in steps of 20%) of the dataset size. The number of features to be deleted in each of these vectors is decided independently by drawing a number from $\{1, \dots, \lceil D/3 \rceil\}$. In each step we use different methods to estimate the kernel matrix and measuring the Root Mean Squared-Error between these estimates and the true kernel matrix computed beforehand. In this experiments, the true distribution of the data is unknown, thus, at each step we estimate a GMM comprising three components.

We do not consider CMI in this experiment, as it was consistently outperformed by the other methods in the previous experiment. Instead, we substitute it by the Incomplete-Case k -Nearest-Neighbors Imputation algorithm (ICkNNI) (HULSE; KHOSH-GOFTAAR, 2014), a well-known distance-based imputation algorithm that does not rely on the estimation of a statistical model for the data. The parameters for ICkNNI were implemented as suggested in Hulse e Khoshgoftaar (2014). Results, in terms of average Root Mean Square Error (RMSE), are presented in Table 5. We employed Wilcoxon’s signed-rank test, with a 5% significance level, to verify the statistical significance of the results. The symbols  and  indicate the result of the hypothesis test ( fail to reject, and  reject).

Unsurprisingly, the performance of all methods decrease with the amount of

missing values. Specifically for ESD and EGK, performance deteriorates due to the quality of distribution estimated for the data, which is affected by the amount of missing data.

Again, EGK achieved the best overall results. This fact provides some evidence that the assumptions taken in the formulation of EGK do not affect negatively its performance on real world data. Additionally, it is interesting to verify that even using a statistical model with only three Gaussians, EGK was able to outperform a non-parametric model such as the ICkNNI.

Table 5 – Gaussian kernel estimation on real-world data: average RMSE

	%	ICkNNI	ESD	EGK
MPG	10%	0.016575 ✗	0.014763 ✗	0.01426
	30%	0.027873 ✗	0.02532 ✗	0.021949
	50%	0.035591 ✗	0.030675 ✗	0.027207
FIRE	10%	0.040201 ✓	0.033981 ✓	0.044753
	30%	0.068591 ✓	0.05635 ✓	0.071251
	50%	0.088094 ✓	0.068472 ✓	0.083955
COL	10%	0.010816 ✗	0.0093255 ✗	0.0077406
	30%	0.018743 ✗	0.015251 ✗	0.013154
	50%	0.023485 ✗	0.019241 ✗	0.016712
HAB	10%	0.045808 ✗	0.041623 ✗	0.035796
	30%	0.077246 ✗	0.068928 ✗	0.059605
	50%	0.098103 ✗	0.083429 ✗	0.073179
PID	10%	0.0025677 ✗	0.0019287 ✗	0.0018322
	30%	0.0044262 ✗	0.0031619 ✗	0.0030542
	50%	0.0055869 ✗	0.0037776 ✗	0.0036778
IRIS	10%	0.037567 ✗	0.034068 ✗	0.028645
	30%	0.069988 ✗	0.060438 ✗	0.051269
	50%	0.090736 ✗	0.07312 ✗	0.062649
COMP	10%	0.012263 ✗	0.016336 ✗	0.014708
	30%	0.020423 ✗	0.026614 ✗	0.024129
	50%	0.026896 ✗	0.032401 ✗	0.029939
SLUMP	10%	0.010205 ✓	0.0089769 ✗	0.0082288
	30%	0.017697 ✗	0.014825 ✗	0.013209
	50%	0.022699 ✗	0.021153 ✗	0.019243

3.3 Conclusion

In this chapter, we presented a methodology to estimate the Gaussian Kernel between two feature vectors X_i and X_j when one or both have missing entries. The proposed method takes the expected value of the kernel $k(X_i, X_j)$ as a transform of the squared Euclidean distance $z = \|X_i - X_j\|^2$. In turn, z is modelled as a Gamma-distributed random variable and a procedure is outlined to compute the parameters that govern this distribution.

The proposed strategy, coined EGK was compared against other methods in the literature, outperforming these in artificial and real-world scenarios.

4 EXPECTED EUCLIDEAN DISTANCE

Given two D -dimensional vectors $X_i = (x_{i,1}, \dots, x_{i,D})^T$ and $X_j = (x_{j,1}, \dots, x_{j,D})^T$, the Euclidean distance η between X_i and X_j is given by

$$\eta = z^{1/2} \triangleq \sqrt{\sum_{d=1}^D (x_{i,d} - x_{j,d})^2}, \quad (4.1)$$

where, as in the previous Chapter, z denotes the squared distance between X_i and X_j .

In this Chapter, we present a methodology to estimate η when vectors $X_i, X_j \in \mathcal{X}$ count on one or more missing components. Following the developments of the previous chapter, we assume z is a Gamma-distributed random variable.

4.1 Formulation

Note that η is a random variable as it is a non-negative transform of X_i and X_j . Hence, taking the expected value of η consists in computing:

$$\mathbb{E}[\eta] = \int_0^{+\infty} \eta p(\eta) d\eta. \quad (4.2)$$

Drawing from the previous chapter, we say z follows a Gamma distribution with parameters α and β that can be estimated from the non-central moments of X_i and X_j as described before. It is then sensible to choose the Nakagami (NAKAGAMI, 1960) distribution for η . By definition, since η is the square-root transform of z :

$$\eta \sim \text{Nakagami}(m, \Omega), \quad (4.3)$$

where m and Ω are, respectively, the shape and spread parameters of the Nakagami distribution. Under this setup, the expected value of η is given by:

$$\mathbb{E}[\eta] = \frac{\Gamma(m + \frac{1}{2})}{\Gamma(m)} \left(\frac{\Omega}{m} \right)^{\frac{1}{2}}. \quad (4.4)$$

In turn, using the method-of-moments, the parameters m and Ω can be written as functions of the mean and variance of z according to:

$$m = \frac{\mathbb{E}^2[z]}{\text{Var}[z]}, \quad \Omega = \mathbb{E}[z]. \quad (4.5)$$

Remind $\mathbb{E}[z]$ and $\text{Var}[z]$ can be computed from the non-central moments of X_i and X_j , as described in Chapter 3.

4.2 Experiments and Results

We perform two different experiments to validate our methodology (EED) . In the first, we study how the uncertainty on the estimation of the missing values affects the quality of the distance estimate. In second, we evaluate EED on real-world data. Table 6 summarizes the details of these experiments. The following subsections present and discuss the results.

Table 6 – Overview of the experiments.

Objective		Setup
EX1	Assess the quality of the distance estimation as a function of the uncertainty on the estimation of the missing values	X_i and X_j are drawn from an univariate Normal distribution with known mean and variance, but X_i missing
EX2	Validate the method on real-world data	Different datasets from the UCI repository were employed.

In the first experiment, EED is compared against the CMI and ESD. As in the case of EEK, EED differs from these methods fundamentally in the level in which the estimation problem is cast. The conceptual differences between CMI, ESD and EED are explicitly shown in Eqs. (4.6) to (4.8).

$$\eta^{EED}(X_i, X_j) = \mathbb{E} \left[\sqrt{\|X_i - X_j\|_2^2} \right], \quad (4.6)$$

$$\eta^{ESD}(X_i, X_j) = \sqrt{\mathbb{E}[\|X_i - X_j\|_2^2]}, \quad (4.7)$$

$$\eta^{CMI}(X_i, X_j) = \sqrt{\|\mathbb{E}[X_i] - \mathbb{E}[X_j]\|^2}. \quad (4.8)$$

As stated by Eirola *et al.* (2013), estimating the missing entries before taking the squared euclidean distance tends to underestimate the expected value of this transform. As a consequence, since $\sqrt{\cdot}$ is strictly increasing:

$$\eta^{CMI}(X_i, X_j) \leq \eta^{ESD}(X_i, X_j). \quad (4.9)$$

For the case in which X_i and X_j are univariate and abide to the conditions for z to be Gamma-distributed, a similar statement can be made for EED by applying Jensen's inequality directly to eq. (4.6):

$$\eta^{CMI}(X_i, X_j) \leq \eta^{EED}(X_i, X_j). \quad (4.10)$$

4.2.1 EX1: Univariate Normal data with known parameters

For this experiment, we fix $X_j = 3$, assume $X_i \sim \mathcal{N}(2, \sigma_n^2)$ and estimate η . Since the distribution of X_i is known, there is no need to estimate a model for the data as the true distribution $\mathcal{N}(2, \sigma_n^2)$ is given.

To obtain a benchmark, we compute a Monte Carlo (MC) estimate of η by performing 10^8 draws of X_i from $\mathcal{N}(2, \sigma_n^2)$, taking the Euclidean distance to X_j for each of these draws and then averaging over the computed distances. This is done to obtain an accurate approximation of the expected value the kernel. Based on that, a method is as good as its estimates are similar to the ones obtained via Monte Carlo. Table 7 shows the averaged Euclidean distance computed by each method for different values of σ_n^2 .

Table 7 – Euclidean distance estimates.

σ^2	MC	CMI	ESD	EED
10^{-2}	1	1	1.005	1
10^{-1}	1.0002	1	1.0488	1.0045
10^0	1.1667	1	1.4142	1.1866
10^1	2.6481	1	3.3166	2.6505
10^2	8.019	1	10.0499	8.0188

As in Subsection 3.2.1, note that CMI computes the same approximation regardless of the value of σ_n^2 . This is expected, since the expected value of X_i depends only on μ . On the other hand, as shown in (EIROLA *et al.*, 2013), the variance of the estimates is taken into account in ESD, providing more accurate results than those obtained with CMI.

While results show ESD clearly outperforms CMI, the quality of its approximation degrades as σ_n^2 increases. In contrast to that, EED maintains a steady performance and obtains the best results for all values of σ^2 . Note also that the results presented follow the behaviour described in Eqs. (4.9) and (4.10).

4.2.2 EX2: Experiments on Real-World Data

We evaluate the performance of EED on real-world datasets. For that purpose, six datasets were selected from the UCI Machine Learning Repository (LICHMAN, 2013). Further details on these datasets are available in Table 4.

Thirty similar rounds of experiment were carried. In each of these, the percentage of instances with missing samples was iteratively increased from 10% to 50% (in steps of 20%) of the dataset size. The number of features to be deleted in each of these vectors is decided independently by drawing a number from $\{1, \dots, \lceil D/3 \rceil\}$. In each step we use different methods to estimate the pairwise distance matrix and measuring the RMSE between these estimates and the true distance matrix computed beforehand. In this experiments, the true distribution of the data is unknown, thus, at each step we estimate a GMM, as specified by Mesquita *et al.* (2017) .

As in subsection 3.2.2, we do not consider CMI in this experiment, as it was consistently outperformed by the other methods in the previous experiment. Instead, we substitute it by the ICkNNI, using the parameters suggested in Hulse e Khoshgoftaar (2014). Results, in terms of average RMSE, are presented in Table 8. We employed Wilcoxon’s signed-rank test, with a 5% significance level, to verify the statistical significance of the results. The symbols ✓ and ✗ indicate the result of the hypothesis test (✓ fail to reject, and ✗ reject).

Table 8 – Euclidean distance estimation on real-world data: average RMSE

	%	ICkNNI	ESD	EED
MPG	10%	0.1672 ✓	0.1620 ✗	0.1607
	30%	0.3036 ✗	0.2780 ✗	0.2758
	50%	0.3967 ✗	0.3593 ✗	0.3561
FIRE	10%	0.3239 ✓	0.2635 ✓	0.2644
	30%	0.6018 ✓	0.5278 ✓	0.5354
	50%	0.7218 ✓	0.8219 ✗	0.7973
COL	10%	0.1665 ✗	0.1203 ✗	0.1176
	30%	0.2968 ✗	0.2475 ✗	0.2431
	50%	0.4477 ✗	0.3162 ✗	0.3121
HAB	10%	0.2803 ✗	0.2140 ✗	0.2047
	30%	0.4510 ✗	0.3923 ✗	0.3805
	50%	0.6064 ✗	0.5118 ✗	0.4971
PID	10%	0.2699 ✗	0.2246 ✗	0.2216
	30%	0.4745 ✗	0.3988 ✗	0.3946
	50%	0.6274 ✗	0.5252 ✗	0.5214
IRIS	10%	0.1509 ✗	0.1202 ✗	0.1184
	30%	0.2617 ✗	0.2195 ✗	0.2148
	50%	0.3492 ✗	0.2828 ✗	0.2800
COMP	10%	0.1731 ✓	0.1791 ✗	0.1761
	30%	0.3067 ✗	0.3261 ✗	0.3243
	50%	0.4192 ✓	0.4247 ✗	0.4198
SLUMP	10%	0.2178 ✗	0.1645 ✗	0.1635
	30%	0.3721 ✗	0.2827 ✗	0.2815
	50%	0.5180 ✗	0.3986 ✗	0.3955

With regard to the RMSE performance, we observe that EED outperforms CMI and ESD in all scenarios, i.e., with small and large amount of missing data. As expected, the performance gap between EED and ESD is smaller than the gap between EED and CMI. The hypothesis test indicates significant difference between EED and the other methods.

4.3 Conclusion

In this chapter, we presented a methodology to estimate the Euclidean distance between two feature vectors X_i and X_j when one or both count on missing entries. The proposed method computes the expected value of the squared-root transform $z^{1/2}$ of the random variable $z = \|X_i - X_j\|^2$. As in the previous chapter, we assume z can be modelled with a Gamma distribution. As a consequence, $z^{1/2}$ is Nakagami-distributed and its expected value can be easily obtained from the distribution parameters.

The proposed strategy, coined EED is validated in artificial and real-world scenarios, outperforming other methods in the literature.

5 EPANECHNIKOV KERNEL

Given two D -dimensional vectors $X_i = (x_{i,1}, \dots, x_{i,D})^T$ and $X_j = (x_{j,1}, \dots, x_{j,D})^T$, the Epanechnikov kernel is given by:

$$k(X_i, X_j) \triangleq \left(1 - \frac{\|X_i - X_j\|^2}{l}\right)^p = l^{-p}(l - z)^p, \quad (5.1)$$

where $p \in \mathbb{N} - \{0\}$ and $l \in \mathbb{R}^+$ are kernel hyper-parameters.

In this Chapter, we present a methodology to estimate $k(X_i, X_j)$ when vectors $X_i, X_j \in \mathcal{X}$ count on one or more missing components. As in the previous Chapters, it is assumed $z \sim \text{Gamma}(\alpha, \beta)$.

5.1 Formulation

Note that the Epanechnikov kernel $k(X_i, X_j)$ is a p -th order polynomial of z and can be expanded to yield:

$$k(X_i, X_j) = \sum_{r=0}^p \binom{p}{r} (l)^{-r} (-z)^r, \quad (5.2)$$

consequently, due to the linearity of expectation, estimating $k(X_i, X_j)$ resumes to computing:

$$\mathbb{E}[k(X_i, X_j)] = \sum_{r=0}^p \binom{p}{r} (-l)^{-r} \mathbb{E}[z^r], \quad (5.3)$$

which is a weighted sum of the non-central moments of z . As in the previous chapters, we assume $z \sim \text{Gamma}(\alpha, \beta)$. Therefore, its i -th non-central moment $\mathbb{E}[z^i]$ is given by:

$$\mathbb{E}[z^i] = \beta^{-i} \frac{\Gamma(\alpha + i)}{\Gamma(\alpha)}, \quad (5.4)$$

and, since i is a non-negative integer, eq. (5.4) simplifies to:

$$\mathbb{E}[z^i] = \beta^{-i} \prod_{j=0}^{i-1} (\alpha + j). \quad (5.5)$$

As before, the parameters α and β of the Gamma distribution can be estimated via the method-of-moments - see Eq. (3.10) - from $\mathbb{E}[z]$ and $\text{Var}[z]$. Also as stated in Chapter 5, $\mathbb{E}[z]$ and $\text{Var}[z]$ can be computed from the non-central moments of X_i and X_j - see Eqs. 3.12 and 3.14.

5.2 Experiments and Results

We perform two different experiments to validate our methodology (EEK) . In the first, we study how the uncertainty on the estimation of the missing values affects the quality of the kernel estimate. In second, we evaluate EEK on real-world data. Table 9 summarizes the details of these experiments. The following subsections present and discuss the results.

Table 9 – Overview of the experiments.

	Objective	Setup
EX1	Assess the quality of the kernel estimation as a function of the uncertainty on the estimation of the missing values	X_i and X_j are drawn from an univariate Normal distribution with known mean and variance, but X_i missing
EX2	Validate the method on Real-world data	Different datasets from the UCI repository were employed.

In the first experiment, EEK is compared against the CMI, and ESD. As in the case of methodologies proposed in previous chapters, EEK differs from these methods fundamentally in the level in which the estimation problem is cast. The conceptual differences between CMI, ESD and EEK are explicitly shown in Eqs. (4.6) to (4.8).

$$\hat{k}_{EEK}(X_i, X_j) = \mathbb{E} \left[\sum_{r=0}^p \binom{p}{r} (-l)^{-r} \|X_i - X_j\|^{2r} \right], \quad (5.6)$$

$$\hat{k}_{ESD}(X_i, X_j) = \sum_{r=0}^p \binom{p}{r} (-l)^{-r} \mathbb{E} [\|X_i - X_j\|^2]^r, \quad (5.7)$$

$$\hat{k}_{CMI}(X_i, X_j) = \sum_{r=0}^p \binom{p}{r} (-l)^{-r} \|\mathbb{E}[X_i] - \mathbb{E}[X_j]\|^{2r}. \quad (5.8)$$

For the case in which X_i and X_j abide to the conditions that make z a Gamma-distributed random variable, when p is such that $g(\nu) = \nu^p$ is convex, applying Jensen's inequality directly to Eq. (5.6) we obtain:

$$\hat{k}_{EEK}(X_i, X_j) \geq \hat{k}_{ESD}(X_i, X_j). \quad (5.9)$$

5.2.1 EX1: Univariate Normal data with known parameters

For this experiment, we set the kernel hyperparameters $p = 2$ and $l = 40$. Furthermore, we fix $X_j = 3$ and assume $X_i \sim \mathcal{N}(2, \sigma_n^2)$. Since the distribution of X_i is known, there is no need to estimate a model for the data as the true distribution $\mathcal{N}(2, \sigma_n^2)$ is given.

To obtain a benchmark, we compute a Monte Carlo (MC) estimate of the kernel by performing 10^8 draws of X_i from $\mathcal{N}(2, \sigma_n^2)$, taking the Euclidean distance to X_j for each of these draws and then averaging over the computed distances. This is done to obtain an accurate approximation of the expected value of the kernel. Based on that, a method is as good as its estimates are similar to the ones obtained via Monte Carlo. Table 10 shows the average Epanechnikov kernel computed by each method for different values of σ_n^2 .

Table 10 – Epanechnikov kernel estimates.

σ^2	MC	CMI	ESD	EEK
10^{-2}	0.9799	0.9801	0.9799	0.9799
10^{-1}	0.9782	0.9801	0.9781	0.9782
10^0	0.9610	0.9801	0.9604	0.9610
10^1	0.8161	0.9801	0.7921	0.8161
10^2	2.0403	0.9801	0.0001	2.0401

Observe that CMI computes the same approximation regardless of the value of σ_n^2 . This is expected, since the expected value of X_i depends only on μ . While both CMI and ESD quickly deteriorate as σ_n^2 increases, EEK is consistent, approximating the MC estimate more accurately. Note that results above presented respect Eq. (5.9).





5.2.2 EX2: Experiments on Real-World Data

We evaluate the performance of EEK on real-world datasets. For that purpose, five datasets were selected from the UCI Machine Learning Repository (LICHMAN, 2013). Further details on these datasets are available in Table 11.

Table 11 – Data sets description

Dataset	Size	Features
IRIS	150	4
HAYES	160	3
HABERMAN (HAB)	306	3
BOSTON STOCKS (STOCK)	950	9
CONCRETE COMPRESSION (COMP)	1030	80

Thirty similar rounds of experiment were carried out. In each of these, the percentage of instances with missing samples was iteratively increased from 10% to 50% (in steps of 20%) of the dataset size. The number of features to be deleted in each of these vectors is decided independently by drawing a number from $\{1, \dots, \lceil D/3 \rceil\}$. In each step we use different methods to estimate the kernel matrix and measuring the RMSE between these estimates and the true kernel matrix computed beforehand. The process was repeated for each $p \in \{2, 3, 4\}$. In this experiments, the true distribution of the data is unknown, thus, at each step we estimate a GMM comprising three components.

As in Subsection 3.2.2, we do not consider CMI in this experiment, as it was consistently outperformed by the other methods in the previous experiment. Instead, we substitute it by the ICkNNI, using the parameters suggested in Hulse e Khoshgoftaar (2014). Results, in terms of average RMSE, are presented in Table 12. We employed Wilcoxon’s signed-rank test, with a 5% significance level, to verify the statistical significance of the results. The symbols  and  indicate the result of the hypothesis test ( fail to reject, and  reject).

With regard to the RMSE performance, we observe that EEK outperforms CMI and ESD in most scenarios, i.e., with small and large amounts of missing data. As expected, the performance gap between EEK and ESD is smaller than the gap between EEK and CMI. The hypothesis test indicates, in most cases, significant difference between EEK and the other methods.

5.3 Conclusion

In this Chapter, we presented a methodology to estimate the Epanechnikov kernel between two feature vectors X_i and X_j when one or both count on missing entries. The proposed method computes the expected value of the kernel as a transform of the random variable $z = \|X_i - X_j\|^2$. As in the previous chapters, we assume z can be modelled with a Gamma distribution. As a consequence, $\mathbb{E}[k(X_i, X_j)]$ becomes a weighted sum of first p non-central moments of z and can be easily obtained from the distribution parameters.

The proposed strategy, EEK, is validated in artificial and real-world scenarios, outperforming other methods in the literature.

Table 12 – Comparison between EEK and other methods.

$p = 2$					
	%	ICkNNI	CMI	ESD	EEK
IRIS	10%	0.3624 ✗	0.3627 ✗	0.3620 ✗	0.3619
	30%	0.3627 ✗	0.3628 ✗	0.3608 ✗	0.3604
	50%	0.3595 ✗	0.3595 ✗	0.3563 ✗	0.3556
HAYES	10%	0.2533 ✗	0.2531 ✗	0.2526 ✗	0.2523
	30%	0.2551 ✗	0.2544 ✗	0.2498 ✗	0.2495
	50%	0.2540 ✗	0.2551 ✗	0.2481 ✗	0.2473
HAB	10%	0.2770 ✗	0.2771 ✗	0.2754 ✗	0.2751
	30%	0.2718 ✗	0.2723 ✗	0.2675 ✗	0.2669
	50%	0.2660 ✗	0.2661 ✗	0.2584 ✗	0.2573
STOCK	10%	0.4444 ✗	0.4447 ✗	0.4426 ✗	0.4423
	30%	0.4452 ✗	0.4458 ✗	0.4400 ✗	0.4390
	50%	0.4462 ✗	0.4473 ✗	0.4369 ✗	0.4353
COMP	10%	0.3546 ✗	0.3556 ✗	0.3513 ✗	0.3504
	30%	0.3553 ✗	0.3580 ✗	0.3446 ✗	0.3427
	50%	0.3542 ✗	0.3601 ✗	0.3363 ✗	0.3333
$p = 3$					
	%	ICkNNI	CMI	ESD	EEK
IRIS	10%	0.4195 ✗	0.4197 ✗	0.4185 ✗	0.4183
	30%	0.4204 ✗	0.4205 ✗	0.4169 ✗	0.4162
	50%	0.4177 ✗	0.4177 ✗	0.4121 ✗	0.4110
HAYES	10%	0.3093 ✗	0.3091 ✗	0.3079 ✗	0.3075
	30%	0.3124 ✗	0.3115 ✗	0.3043 ✗	0.3037
	50%	0.3121 ✗	0.3137 ✗	0.3023 ✗	0.3010
HAB	10%	0.3315 ✗	0.3317 ✗	0.3290 ✗	0.3290
	30%	0.3256 ✗	0.3264 ✗	0.3191 ✗	0.3187
	50%	0.3196 ✗	0.3200 ✗	0.3081 ✗	0.3073
STOCK	10%	0.4392 ✗	0.4397 ✗	0.4368 ✗	0.4346
	30%	0.4412 ✗	0.4426 ✗	0.4342 ✗	0.4336
	50%	0.4425 ✗	0.4449 ✗	0.4302 ✗	0.3292
COMP	10%	0.3536 ✗	0.3551 ✗	0.3493 ✓	0.3495
	30%	0.3558 ✗	0.3603 ✗	0.3423 ✗	0.3418
	50%	0.3560 ✗	0.3645 ✗	0.3325 ✗	0.3312
$p = 4$					
	%	ICkNNI	CMI	ESD	EEK
IRIS	10%	0.3624 ✗	0.3627 ✗	0.3620 ✗	0.3619
	30%	0.3627 ✗	0.3628 ✗	0.3608 ✗	0.3604
	50%	0.3595 ✗	0.3595 ✗	0.3563 ✗	0.3556
HAYES	10%	0.3437 ✗	0.3436 ✗	0.3415 ✗	0.3411
	30%	0.3480 ✗	0.3470 ✗	0.3372 ✗	0.3366
	50%	0.3486 ✗	0.3506 ✗	0.3348 ✗	0.3335
HAB	10%	0.3602 ✗	0.3602 ✗	0.3569 ✓	0.3568
	30%	0.3555 ✗	0.3567 ✗	0.3467 ✗	0.3460
	50%	0.3507 ✗	0.3514 ✗	0.3353 ✗	0.3337
STOCK	10%	0.4221 ✗	0.4227 ✗	0.4192 ✗	0.4191
	30%	0.4248 ✗	0.4266 ✗	0.4165 ✗	0.4162
	50%	0.4261 ✗	0.4293 ✗	0.4116 ✗	0.4111
COMP	10%	0.3321 ✗	0.3336 ✗	0.3269 ✓	0.3272
	30%	0.3342 ✗	0.3396 ✗	0.3187 ✗	0.3182
	50%	0.3373 ✗	0.3471 ✗	0.3103 ✗	0.3089

6 EXPECTED VALUE OF BASIS FUNCTIONS

Single-Layer Feedforward Neural Networks (SLFNNs) can often be expressed in terms of basis expansion functions, *i.e.*, the predicted output $\hat{y} \in \mathbb{R}$ for an input vector $X \in \mathbb{R}^D$ can be expressed as

$$\hat{y} = \sum_{h=1}^H \kappa_h \phi_h(X). \quad (6.1)$$

where $\phi_h : \mathbb{R}^D \rightarrow \mathbb{R}$ is the activation function of the h -th hidden neuron and $\kappa_h \in \mathbb{R}$ is the weight of the link between this neuron and the output node.

For instance, for conventional Random Neural Networks (RNNs) using the sigmoid (also known as logistic) activation function, we have:

$$\hat{y} = \sum_{h=1}^H \kappa_h g(\lambda_h \cdot X) \quad (6.2)$$

where $g(t) = 1/(1 + e^{-t})$ and $\lambda_h \in \mathbb{R}^D$ is the vector of weights that links the input layer to the h -th hidden neuron. Thus, ϕ_h can be written as a non-linear transform of $\lambda_h \cdot X$. This is also true - with different expressions for $g(\cdot)$ - for Single-Layer Feedforward Neural Network (SLFNN)s using the hyperbolic tangent, logit, probit or cosine as the activation function.

On the other-hand, for centroid-based SLFNNs, such as Radial Basis Function Networks and q -Generalized Random Neural Networks, Eq. (6.1) is equivalent to:

$$\hat{y} = \sum_{h=1}^H \kappa_h g(\|X - \lambda_h\|^2) \quad (6.3)$$

with $\lambda \in \mathbb{R}^D$ now as the h -th centroid of the network.

In this chapter, we propose two sampling strategies to estimate the value of a basis expansion function $\phi(X)$ when X has missing entries. The first one addresses the case in which ϕ can be expressed as a transform of $\lambda \cdot X$. The latter deals with the case in which $\phi(X)$ is a transform of $\|X - \lambda\|^2$. Both strategies are based on the Unscented Transform (UT) and sample only $O(1)$ scalar points. Special attention is given to the sigmoid and the q -Gaussian activation functions.

6.1 Formulation

The problem of estimating $\phi(X)$ for a D -dimensional vector X counting on missing entries consists in computing

$$\mathbb{E}[\phi(X)] = \int_{\mathbb{R}^D} \phi(X)p(X)dX, \quad (6.4)$$

for which there is no trivial general solution and tailored ones depend on both the format of $\phi(\cdot)$ and $p(\cdot)$. However, for any $\phi(\cdot)$ and $p(\cdot)$, it is possible to approximate Eq. (6.4) via sampling or numerical integration methods.

The UT, originally proposed in (JULIER; UHLMANN, 1997), is a sampling-based method for estimating statistical moments of a probability distribution associated to a random variable which results from a nonlinear transformation of another random variable (LEÃO; YONEYAMA, 2011).

In order to estimate $\phi(X)$ using the UT, a set $\mathcal{S} = \{\gamma_l\}_{l=1}^L \subset \mathbb{R}^D$ of sigma points (SPs), with respective weights $\{k_l\}_{l=1}^L \subset \mathbb{R}$, associated to the original random variable X are deterministically chosen. Then, the SPs are passed through $\phi(\cdot)$, resulting in a transformed set of SPs. Finally, the transformed SPs (and their corresponding weights) are used in order to approximate $\mathbb{E}[\phi(X)]$. Although there is no restriction on their sign, the weights k_1, \dots, k_L must respect the convexity constraint

$$\sum_{l=1}^L k_l = 1 \quad (6.5)$$

to provide an unbiased estimate (JULIER; UHLMANN, 2004).

The implementation of the UT to estimate $\phi(X)$ can then be summarized by the following equations:

$$\delta_l \leftarrow \phi(\gamma_l) \quad \forall 1 \leq l \leq L, \quad (6.6)$$

$$\mathbb{E}[\phi(X)] \approx \sum_{l=1}^L k_l \delta_l. \quad (6.7)$$

There are different possible ways to choose the SPs and respective weights. Let M denote the set of indices corespoding to missing features in X . A common approach is to use a symmetric set of $L = 2|M| + 1$ SPs with identical weights, as described in

Equations (6.8) to (6.11).

$$\gamma_1 = \mathbb{E}[X] \quad (6.8)$$

$$\gamma_l = \gamma_1 + \left[\sqrt{|M| \Sigma} \right]_{l-1} \quad \forall 1 < l \leq |M| + 1 \quad (6.9)$$

$$\gamma_l = \gamma_1 - \left[\sqrt{|M| \Sigma} \right]_{l-(d+1)} \quad \forall d + 1 < l \leq 2|M| + 1 \quad (6.10)$$

$$k_l = \frac{1}{2|M| + 1} \quad \forall 1 \leq l \leq 2|M| + 1 \quad (6.11)$$

where $\left[\sqrt{|M| \Sigma} \right]_l$ denotes the l -th row of the matrix square root of $|M| \Sigma$, which is the covariance matrix Σ of X_M (conditioned on X_O) multiplied by the number of missing entries $|M|$.

Although this UT approach could be applied directly to estimate the value of arbitrary basis expansion functions in our context, the number of required samples L grows with the number of missing entries, which could make it computationally inefficient when X counts on many missing features.

To alleviate this problem, we propose two methodologies based on the UT that require only three one-dimensional sigma points, independent of $|M|$. The first one, presented in Subsection 6.1.1, is tailored to the sigmoid function and can be easily generalized to any $\phi(\cdot)$ that can be expressed as transform of $\lambda \cdot X$. The latter, in Subsection 6.1.2, deals with the q -Gaussian function and can be adapted for any $\phi(\cdot)$ that is a transform of $\|X - \lambda\|^2$.

6.1.1 Sigmoid Function

Given an input vector $X = (x_1, \dots, x_D)^T$, the sigmoid function is given by:

$$f_\sigma(X) = \frac{1}{1 + e^{-\lambda \cdot X}}, \quad (6.12)$$

in which $\lambda = (\lambda_1, \dots, \lambda_D)^T$ is a predefined constant vector.

Note that $f_\sigma(X)$ can be written as a transform of the random variable $\lambda \cdot X$, whose expectation is given by:

$$\mathbb{E}[\lambda \cdot X] = \sum_{d=1}^D \lambda_d \mathbb{E}[x_d], \quad (6.13)$$

and has variance:

$$\text{Var}[\lambda \cdot X] = \sum_{d=1}^D \lambda_d^2 \text{Var}[x_d]. \quad (6.14)$$

Using the aforementioned UT scheme, we can approximate $\mathbb{E}[f_\sigma(X)]$ using $L = 3$ sigma points as follows:

$$\mathbb{E}[f_\sigma(X)] \approx \sum_{i \in \{-1, 0, 1\}} \frac{(1 + \exp\{-\mathbb{E}[\lambda \cdot X] - i\text{Var}[\lambda \cdot X]\})^{-1}}{3} \quad (6.15)$$

It is important to notice this methodology also applies to any transform of X that can be written as a function of $\lambda \cdot X$.

6.1.2 *q-Gaussian Function*

For an input vector $X = (x_1, \dots, x_D)^T$, the q -Gaussian activation function can be expressed as:

$$G(X) = e_q(-\|X - \lambda\|^2 \nu^{-1}) \quad (6.16)$$

where $\lambda = (\lambda_1, \dots, \lambda_D)^T$, $\nu > 0$ and $q \in \mathbb{R}$ are predefined constants while

$$e_q(t) = [1 + (1 - q)t]^{\frac{1}{1-q}}. \quad (6.17)$$

Note that $G(X)$ can be written as a transform of $\|X - \lambda\|^2$, whose expectation is given by

$$\mathbb{E}[\|X - \lambda\|^2] = \sum_{d=1}^D (\mathbb{E}[x_d] - \lambda_d)^2 + \text{Var}[x_d], \quad (6.18)$$

and has variance:

$$\text{Var}[\|X - \lambda\|^2] = \sum_{d=1}^D \mathbb{E}[x_d^4] - \mathbb{E}[x_d^2]^2 + 4\lambda_d^2 \text{Var}[x_d]. \quad (6.19)$$

Thus, $\mathbb{E}[G(X)]$ can be approximated using the aforementioned UT scheme with exactly $L = 3$ sigma points, as follows:

$$\mathbb{E}[G(X)] \approx \sum_{i \in \{-1, 0, 1\}} \frac{e_q(-(\mathbb{E}[\|X - \lambda\|^2] + i\sqrt{\text{Var}[\|X - \lambda\|^2]})\nu^{-1})}{3} \quad (6.20)$$

Notice that this methodology can be trivially adapted to estimate the value of any specific transform $\phi(X)$ that can be expressed as a function of $\|X - \lambda\|^2$.

6.2 Experiments and Results

We perform two different sets of experiments to validate our methodologies. In the first, we study how the uncertainty on the estimation of the missing values affects the

quality of the function estimates. In the second, we validate our estimation procedures on real-world data. Table 13 summarizes the details of these experiments. The following subsections present and discuss the results. In the first experiment, we compare our strategy against the CMI. In the later ones, we also compare it against ICkNNI.

Table 13 – Overview of the experiments.

Objective		Setup
EX1	Assess the quality of the estimation as a function of the uncertainty on the estimation of the missing values	λ is fixed and X is drawn from an univariate Normal distribution with known mean and variance.
EX2	Validate the method on Real-world data	Different datasets from the UCI repository were employed.

6.2.1 EX1: Univariate Normal data with known parameters

For this experiment, we set $\lambda = 3$ and assume $X \sim \mathcal{N}(2, \sigma_n^2)$. Since the distribution of X is known, there is no need to estimate a model for the data as the true distribution $\mathcal{N}(2, \sigma_n^2)$ is given. For the q -Gaussian function, we set $q = 1/2$ and $\nu = 4$.

To obtain benchmarks, we compute Monte Carlo (MC) estimates of both f_σ and the q -Gaussian by performing 10^8 draws of X from $\mathcal{N}(2, \sigma_n^2)$, taking the value of the transforms for each sample and then average over the respective computed values. This is done to obtain an accurate approximation of the expected value of the basis functions. Based on that, a method is as good as its estimates are similar to the ones obtained via Monte Carlo.

We compare our methodologies against CMI, a common imputation procedure. Table 14 shows the average value of sigmoid function computed by each method for different values of σ_n^2 . Table 15 holds similar statistics for the q -Gaussian function. In these tables, both our methodologies are referred to as Simplified UT (SUT).

Table 14 – Sigmoid function estimates.

σ_n^2	MC	CMI	SUT
10^{-2}	0.9974	0.9975	0.9975
10^{-1}	0.9963	0.9975	0.9967
10^0	0.9541	0.9975	0.9833
10^1	0.7402	0.9975	0.6757
10^2	0.5972	0.9975	0.6658

Table 15 – q -Gaussian function estimates.

σ_n^2	MC	CMI	SUT
10^{-2}	0.9381	0.9385	0.9379
10^{-1}	0.9332	0.9385	0.9327
10^0	0.8858	0.9385	0.8828
10^1	0.6599	0.9385	0.5869
10^2	17.9307	0.9385	17.9307

Unsurprisingly, the estimates obtained using CMI are the same regardless of the value of σ_n^2 . This is expected, since the the expected value of X depends only of μ . While both CMI quickly deteriorate as σ_n^2 increases, SUT is consistent, approximating the MC estimate more accurately.

6.2.2 EX2: Experiments on Real-World Data

We evaluate the performance of the proposed methodology in real-world datasets. For that purpose, 7 datasets were selected from the UCI Machine Learning Repository (LICHMAN, 2013). Further details on these datasets are available in Table 16.

Table 16 – Data sets description

Dataset	Size	Features
CANCER	194	32
MPG	392	7
CPU	209	9
CONCRETE COMPRESSION (COMP)	1030	80
BOSTON HOUSING (HOUSING)	506	13
RED WINE (RED)	1599	11
WHITE WINE (WHITE)	4898	3

For both the sigmoid and q -Gaussian functions, twenty similar rounds of experiment were carried. In each of these rounds, the percentage of instances with missing samples was iteratively increased from 10% to 50% (in steps of 20%) of the dataset size.

The number of features to be deleted in each of these samples is decided independently by drawing a number from $\{1, \dots, \lceil D/3 \rceil\}$. In each step we use different methods to estimate the transforms, and measured the RMSE between the obtained estimates and the true function values computed beforehand. At each these, the vector λ , in each iteration, was chosen at random, independently and with equal probability, from the complete examples in the dataset. Since the true distribution of the data is unknown, we estimate a GMM comprising three components. Besides CMI, we also compare our method against ICkNNI, using the parameters suggested in Hulse e Khoshgoftaar (2014).

Results, in terms of average RMSE, are presented in Table 17 for the sigmoid function and in Table 18 for the q -gaussian. We employed Wilcoxon’s signed-rank test, with a 5% significance level, to verify the statistical significance of the results. The symbols \checkmark and \times indicate the result of the hypothesis test (\checkmark fail to reject, and \times reject).

Table 17 – Comparison between SUT and other methods to compute the sigmoid function on real-world data - RMSE values.

	%	ICkNNI	CMI	SUT
Cancer	10	66.537 \times	65.94 \times	54.4
	30	64.953 \times	68.148 \times	46.746
	50	70.419 \times	69.592 \times	43.209
MPG	10	3.7966 \times	3.7682 \times	2.8901
	30	4.1204 \times	3.8626 \times	2.9708
	50	4.2425 \times	4.0861 \times	3.0207
CPU	10	123.43 \times	120.58 \times	82.31
	30	106.78 \times	103.69 \times	78.437
	50	115.84 \times	120.27 \times	78.753
Compression	10	8.2138 \times	8.1956 \times	8.0523
	30	8.4074 \checkmark	8.4325 \checkmark	8.3549
	50	8.5655 \checkmark	8.5391 \checkmark	8.4992
Boston Housing	10	5.0635 \times	5.0479 \times	4.625
	30	5.2971 \times	5.2317 \times	4.6414
	50	5.5568 \times	5.3776 \times	4.7814
Red Wine	10	0.67824 \times	0.67898 \times	0.66255
	30	0.68191 \times	0.68054 \times	0.65806
	50	0.68004 \times	0.68107 \times	0.65715
White Wine	10	0.75253 \times	0.7523 \times	0.74851
	30	0.75279 \times	0.75293 \times	0.74967
	50	0.75403 \times	0.75308 \checkmark	0.7517

Table 18 – Comparison between SUT and other methods to compute the q -Gaussian function on real-world data - RMSE values.

	%	ICkNNI	CMI	SUT
Cancer	10	61.774✗	62.8✗	45.119
	30	62.731✗	70.854✗	42.632
	50	74.657✗	63.304✗	40.526
MPG	10	2.9974✗	3.0086✗	2.83
	30	3.019✗	3.0006✗	2.8676
	50	3.1313✗	3.1521✗	2.9199
CPU	10	52.253✗	55.021 ✓	87.105
	30	110.5 ✓	112.11 ✓	83.031
	50	187.92✗	154.29✗	70.111
Compression	10	7.7274 ✓	7.7412 ✓	7.7099
	30	7.8419 ✓	7.8907 ✓	7.8701
	50	7.964 ✓	7.9434 ✓	7.9131
Boston Housing	10	4.6463✗	4.577✗	3.9617
	30	4.983✗	5.3899✗	4.2295
	50	5.3723✗	5.206✗	4.3628
Red Wine	10	0.68799✗	0.68827✗	0.65382
	30	0.68813✗	0.68912✗	0.64616
	50	0.6918✗	0.6991✗	0.64542
White Wine	10	0.78623 ✓	0.77484 ✓	0.75504
	30	0.75943 ✓	0.79594 ✓	0.73886
	50	0.7798 ✓	0.84248 ✓	0.74111

As one can notice, SUT achieved the best overall results in most of the cases. Additionally, it is interesting to verify that even using a statistical model with only three Gaussians, SUT was able to outperform a non-parametric model such as the ICkNNI.

6.3 Conclusion

In this chapter, we presented methodologies to estimate the value of basis functions from incomplete feature vectors. The proposed strategies used the unscented transform to compute the expected value of the transforms. It is important to highlight that our strategies require $O(1)$ samples, more specifically three, independent of the number of missing entries on the input vector.

The proposed strategies were validated in artificial and real-world scenarios, outperforming other methods in the literature.

7 CONCLUDING REMARKS

The contribution presented in this thesis is fourfold and consists on methodologies to estimate the value of the Gaussian Kernel, the Euclidean Distance, the Epanechnikov kernel and of arbitrary basis functions in the presence of missing data. These contributions are presented, respectively, in Chapters 3, 4, 5, 6. The common idea behind all the solutions is to treat the missing entries in the feature vectors as random variables and taking the expected value of the quantities of interest.

In Chapters 3, 4 and 5, the expectations are taken with respect to the (Gamma-distributed) squared distance z between these vectors. By doing so, we provide elegant formulas which depend only on the non-central moments of the missing components.

In Chapter 6, we provide tools for estimating the value of arbitrary basis functions. The proposed estimation procedure is based on the Unscented Transform and requires $O(1)$ samples, independent of the number of missing entries on the input.

The proposed methodologies take into account the uncertain nature of these entries, which would otherwise be lost if data was directly imputed.

To obtain estimates of the non-central moments needed, we assume the data distribution can be modelled as a GMM, whose parameters can be computed via EM. It is important to highlight that the developments presented are not bound to this assumption as the GMM could be easily substituted by any other probabilistic model.

The proposed approaches were validated against standard strategies to handle missing data, ESD (EIROLA *et al.*, 2013; EIROLA *et al.*, 2014) and ICkNNI (HULSE; KHOSHGOFTAAR, 2014), which, in contrast to the former ones, is non-parametric and does not rely on the quality of a model estimated from the data. Even though our methodologies also depend on the quality of such a model, they consistently outperformed the aforementioned methods in both artificial and real-world scenarios.

An obvious unfolding of this work is to apply the methods presented in the previous chapters directly to Machine Learning algorithms. This front has already been partially explored in Mesquita *et al.* (2017).

REFERENCES

- ACUÑA, E.; RODRIGUEZ, C. **The Treatment of Missing Values and its Effect on Classifier Accuracy**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. 639–647 p.
- ASTE, M.; BONINSEGNA, M.; FRENO, A.; TRENTIN, E. Techniques for dealing with incomplete data: a tutorial and survey. **Pattern Analysis and Applications**, v. 18, n. 1, p. 1–29, 2015.
- AZZALINI, A. A class of distributions which includes the normal ones. **Scandinavian Journal of Statistics**, v. 12, n. 2, p. 171–178, 1985.
- COVO, S.; ELALOUF, A. A novel single-gamma approximation to the sum of independent gamma variables, and a generalization to infinitely divisible distributions. **Electron. J. Statist.**, The Institute of Mathematical Statistics and the Bernoulli Society, v. 8, n. 1, p. 894–926, 2014.
- EIROLA, E.; DOQUIRE, G.; VERLEYSEN, M.; LENDASSE, A. Distance estimation in numerical data sets with missing values. **Information Sciences**, v. 240, p. 115 – 128, 2013. ISSN 0020-0255.
- EIROLA, E.; LENDASSE, A.; VANDEWALLE, V.; BIERNACKI, C. Mixture of gaussians for distance estimation with missing data. **Neurocomputing**, v. 131, p. 32 – 42, 2014. ISSN 0925-2312.
- GENTON, M. G. **Skew-Elliptical Distributions and Their Applications: A Journey Beyond Normality**. [S.l.]: Chapman and Hall/CRC, 2004.
- GHEYAS, I. A.; SMITH, L. S. A neural network-based framework for the reconstruction of incomplete data sets. **Neurocomputing**, v. 73, n. 16–18, p. 3039 – 3065, 2010.
- HULSE, J. V.; KHOSHGOFTAAR, T. M. Incomplete-case nearest neighbor imputation in software measurement data. **Information Sciences**, v. 259, p. 596 – 610, 2014. ISSN 0020-0255.
- HUNT, L.; JORGENSEN, M. Mixture model clustering for mixed data with missing information. **Comput. Stat. Data Anal.**, v. 41, n. 3-4, p. 429–440, jan. 2003.
- JOHNSON N., K. S.; BALAKRISHNAN, N. **Continuous univariate distributions**. [S.l.]: Wiley, 1995. ISBN 9780471584940.
- JULIER, S. J.; UHLMANN, J. K. A new extension of the Kalman filter to nonlinear systems. In: **SPIE Aerosense Symposium**. [S.l.: s.n.], 1997. p. 182–193.
- JULIER, S. J.; UHLMANN, J. K. Unscented filtering and nonlinear estimation. **Proceedings of the IEEE**, v. 92, n. 3, p. 401–422, Mar 2004.
- KANG, P. Locally linear reconstruction based missing value imputation for supervised learning. **Neurocomputing**, v. 118, p. 65 – 78, 2013.
- LEÃO, B. P.; YONEYAMA, T. On the use of the unscented transform for failure prognostics. In: **IEEE Aerospace Conference**. Big Sky: IEEE, 2011.

LICHMAN, M. **UCI Machine Learning Repository**. 2013. Disponível em: <<http://archive.ics.uci.edu/ml>>.

LOBATO, F.; SALES, C.; ARAUJO, I.; TADAIESKY, V.; DIAS, L.; RAMOS, L.; SANTANA, A. Multi-objective genetic algorithm for missing data imputation. **Pattern Recognition Letters**, v. 68, Part 1, p. 126 – 131, 2015.

MESQUITA, D. P.; GOMES, J. P.; JUNIOR, A. H. S.; NOBRE, J. S. Euclidean distance estimation in incomplete datasets. **Neurocomputing**, v. 248, p. 11 – 18, 2017. Neural Networks : Learning Algorithms and Classification Systems.

MOLENBERGHS, G.; FITZMAURICE, G.; KENWARD, M. G.; TSIATIS, A.; VERBEKE, G. **Handbook of missing data methodology**. Hoboken, NJ: CRC Press, 2014. (Chapman Hall/CRC handbooks of modern statistical methods).

NAKAGAMI, M. The m-distribution – A general formula of intensity distribution of rapid fading. In: HOFFMANN, W. C. (Ed.). **Statistical Methods in Radio Wave Propagation**. Elmsford, NY: [s.n.], 1960.

ROBERTS, C.; GEISSER, S. A necessary and sufficient condition for the square of a random variable to be gamma. **Biometrika Trust**, v. 53, n. 1/2, p. 275–278, jun. 1966.

SOVILJ, D.; EIROLA, E.; MICHE, Y.; BJÖRK, K.-M.; NIAN, R.; AKUSOK, A.; LENDASSE, A. Extreme learning machine for missing data using multiple imputations. **Neurocomputing**, v. 174, Part A, p. 220 – 231, 2016.