Technical University of Catalonia

Barcelona School of Informatics

Study and extension of kernel functions for missing value treatment

Author:
Yaroslav Hernández Potiomkin

Supervisor:
Lluís A. Belanche Muñoz

January 20, 2015

Contents

1 Introduction

Modern modelling problems are difficult for a number of reasons, including the challenge of dealing with a significant amount of missing information. Kernel methods have won great popularity as a reliable machine learning tool; in particular, Support Vector Machines (SVMs) are kernel-based methods that are used for tasks such as classification and regression, among others [13]. The kernel function is a very flexible container under which to express knowledge about the problem as well as to capture the meaningful relations in input space. Some classical modelling methods –like Naïve Bayes and CART decision trees– are able to deal with missing values without any preprocessing step. However, the process of optimizing an SVM assumes that the training data set is complete.

The main goal of this work is to study kernel functions and extend them in such a way that the kernel-based methods, mentioned above, are able to handle data sets with missing values directly. There is a very big range of possibilities when talking about types of kernel functions. We had selected the most important ones, as they are widely used and by this reason their extension may become a very useful and practical approach for the missing value problem.

The extensions are then compared with other methods through statistical analysis and observing their behaviour ob the different scenarios, for instance, varying the rate of missing information or examining their performance on a real data set problem.

The data sets with missing information appear in many applications and studies, as biological indicators or social source surveys and many others. In addition, when working with kernel methods, these do not accept incomplete data sets and thus, an imputation procedure is then necessary but often leads to a very complex solutions due to the missingness mechanism that will be discussed later. So, an extension to kernel functions becomes a very interesting alternative.

This study is based on previous work [2, 5] which handles missing values in kernel methods by extending kernel functions. There are several methods that use kernel functions, but here the learning method corresponds to Support Vector Machines (SVMs).

The work had been divided in several parts:

- Study of related literature and knowledge acquisition
- Analytical kernel function extension
- Implementation
- Experimentation
- Analysis of the results

Several SVM kernel functions and KDE basis functions had been proposed for the analysis. After the extension, and careful implementation, the methods had been compared, including imputation models. Finally, several results had been presented and future work discussion.

2 Planification, methodology and alternatives

The planification have been followed as strict as possible. The working phase consisted in several iterative steps, composed by

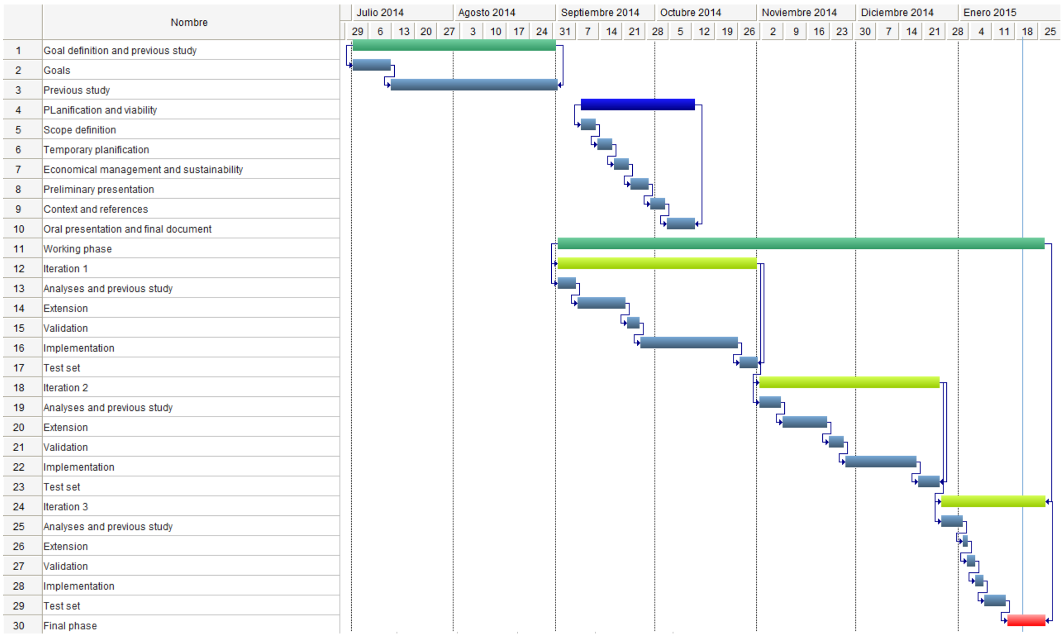
- Analyses
- Extension
- Validation
- Implementation
- Experimentation

The duration of each iteration has been incremented from the beginning of the semester due to:

Analytical mistakes detected during the implementation Computational efficiency

Finally, the main goals were achieved. The most important kernel functions were analyzed and extended.

In the following figure can be seen the final Gantt diagram that matches the whole work performed during the realization of this sudy.



	Nombre
1	Goal definition and previous study
2	Goals
3	Previous study
4	PLanification and viability
5	Scope definition
6	Temporary planification
7	Economical management and sustainability
8	Preliminary presentation
9	Context and references
10	Oral presentation and final document
11	Working phase
12	Iteration 1
13	Analyses and previous study
14	Extension
15	Validation
16	Implementation
17	Test set
18	Iteration 2
19	Analyses and previous study
20	Extension
21	Validation
22	Implementation
23	Test set
24	Iteration 3
25	Analyses and previous study
26	Extension
27	Validation
28	Implementation
29	Test set
30	Final phase

Figure 1: Final *Gantt* diagram.

Regarding the methodology, the programming languages that have been used are: Maple, Matlab, R and C++. The first two have been used to deal with symbolic expressions and for test purposes. R deals with data and training parts. C++ is used uniquely to compute the Gram matrix (expleined later)

The R scripts perform the following tasks:

- Dataset generation
- Parameter estimation
- Model training
- Presentation of results

Matlab is used for numerical computing and results checking.

Regarding the error detection and analyses of alternatives, the computation performance had to be reviewed and a new programming language was introduced (C++). Additionally, the parameter estimation had to be reviewed.

The choice of the best solution was based on the analyses of the bottleneck in the extended methods computation.

3 Preliminaries

In this chapter we will develop the theoretical foundations of the methods used in this work. First, we will start introducing the SVM method. Later on, we will define and present several kernel functions and finally we will talk about the missing value problem and its solutions.

3.1 Sparse Kernel Machines

The algorithms based on kernel functions that determine *sparse* solution, are such that the evaluation of this kernel functions is done only over a small subset of the training dataset. The parameters of the SVM (*Support Vector Machine*) model are found through the optimization of a quadratic function subject to certain conditions, a set of linear inequations. Thus, the optimum is always found.

Let $D = \{(\boldsymbol{x}_1, t_1), ..., (\boldsymbol{x}_N, t_N)\}$ the dataset to be separated, where $\boldsymbol{x}_n \in \mathbb{R}^d$ and $t_n \in \{-1, 1\}$. Linear model for two classes:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + w_0 \tag{1}$$

Regarding maximum margin separators, the OHS (optimum hiperplane separator) is such that defines the largest margin.

$$\gamma = \max_{\boldsymbol{w}, w_0} \min_{1 \leq n \leq N} m(\boldsymbol{x}_n)$$

where $m(\boldsymbol{x}) = \frac{t(\boldsymbol{w}^T \phi(\boldsymbol{x}) + w_0)}{\|\boldsymbol{w}\|}$, the distance from the point to the separator and $\boldsymbol{w}^T \phi(\boldsymbol{x}) + w_0$ is the separator. Is forced that $\min_{\boldsymbol{x}_n} |\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + w_0| = 1$ with $n = 1 \div N$. The support vectors, by definition, are such that $\boldsymbol{x}_n$ where $t_n(\boldsymbol{w}^T \boldsymbol{x}_n + w_0) = 1$.

Then, the margin to be maximized is $\gamma = \frac{1}{\|\boldsymbol{w}\|}$, that is equivalent to minimize $\frac{\|\boldsymbol{w}\|^2}{2}$; that is to say, the optimization problem is the following:

$$\min_{\boldsymbol{w}, w_0} \frac{\|\boldsymbol{w}\|^2}{2}$$

that, at the same time, is subject to a set of conditions of the form:

$$t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + w_0) \geq 1$$

with $n = 1 \div N$. In order to solve this optimization problem with the corresponding conditions, is convenient to introduce *Lagrange* multipliers of the form $a_n \geq 0$ with $n = 1 \div N$, that results in a *Lagrangian* function:

$$L(\boldsymbol{w}, w_0, \boldsymbol{a}) = \frac{\|\boldsymbol{w}\|^2}{2} - \sum_{n=1}^{N} a_n\{t_n(\boldsymbol{w}^T\phi(\boldsymbol{x}_n) + w_0) - 1\}$$

where the sign $(-)$ is due to the fact that the conditions that been multiplied by (-1), minimizing with respect to $\boldsymbol{w}$ and w_0 and maximizing with respect to $\boldsymbol{a}$. Derivative with respect to $\boldsymbol{w}$ and w_0 is obtained:

$$\boldsymbol{w} = \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x_n})$$

and

$$0 = \sum_{n=1}^{N} a_n t_n$$

So substituiting in the *Lagrangian* function it gives dual representation of the maximum margin separator problem:

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$

where the kernel function is defined as $k(\boldsymbol{x}, \boldsymbol{x'}) = \phi(\boldsymbol{x})^T\phi(\boldsymbol{x'})$.

Substituiting these results in the equation (1) is obtained:

$$y(\boldsymbol{x}) = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x}_n) + w_0$$

Using the support vector properties $\boldsymbol{x}_n$, that satisfy $t_n y(\boldsymbol{x}_n) = 1$ and using the previous expression it gives:

$$t_n\left(\sum_{m\in S} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) + w_0\right) = 1$$

where S is the set of indexes that correspond to support vectors. Multiplying by t_n both sides it results in:

$$w_0 = \frac{1}{N_S}\sum_{n\in S}\left(t_n - \sum_{m\in S} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m)\right)$$

where N_S is the total number of support vectors.

In the practice, when classes distributions overlap it can lead to a bad model generalization. For this reason is convenient to modify support vector machine so that it allows some data points to be missclassified. Therefore *slack* variables are introduced. They representthe penalties for the missclassified points. In this way, the data points that are inside or just over the boundary have $\xi_n = 0$, those that are on the decision boundary $\xi_n = 1$, $\xi_n > 1$ for the points that are missclassified and $\xi_n = |t_n - y(\boldsymbol{x}_n)|$ for the rest.

The following conditions arise:

$$t_n y(\boldsymbol{x}_n) \geq 1 - \xi_n$$

with $\xi_n \geq 0$ and $n = 1 \div N$. This environment is sensitive to the *outliers*, as the penalty for the missclassified points is incremented linerly with ξ. Thus, the objective becomes to maximize the margin and penalize *softly* the points that lie on the wrong side of the boundary of the margin. The expression to be minimized is as follows:

$$C \cdot \sum_{n=1}^{N} \xi_n + \frac{\|\boldsymbol{w}\|^2}{2}$$

where $\sum_{n=1}^{N} \xi_n$ is the upper bound of the number of missclassified points and the parameter $C > 0$ (cost constant) controls the balance between the *slack* variables penalty and the margin; that is to say, establishes the control of minimizing training data error and the model complexity. When $C \to \infty$ the margin decreases and can over-fit, otherwise if $C \to 0$ the margin increases and the generalization is better, but it can lead to under-fitting. Usually, is sorted out by k-fold cross-validation.

The *Lagrangian* form is:

$$L(\boldsymbol{w}, w_0, \boldsymbol{a}) = \frac{\|\boldsymbol{w}\|^2}{2} + C \cdot \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} a_n \{t_n y(\boldsymbol{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N} \mu_n \xi_n$$

The dual representation is obtained in similar way as in the previous case, but without *slack* variables. Actually, the model $y(\boldsymbol{x}_n)$ has the same appearance in both cases, but the *Lagrangian* multipliers, for example, are now subject to $0 \leq a_n \leq C$ with $n = 1 \div N$. Other conditions are added too [12].

3.2 Kernel functions

By definition, a function that returns the inner product between the images of two inputs in some feature space is known as a *kernel function*.

This feature space must be an inner product space; those that satisfy the existence of real-valued symmetric bilinear map that also satisfies $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \geq 0$. Actually, *Hilbert* space satisfies these conditions and also the separability and completeness properties [13].

Another, more intuitive way of seeing the kernel functions is as a similarity measure (inner product) between two inputs expressed in some feature space.

Among the most widely used and well-known Kernels we find:

- Linear Kernel
$$K(u, v) = \langle u, v \rangle$$
- Polynomial Kernel
$$K(u, v) = (\langle u, v \rangle + \gamma)^d$$
with $\gamma \in \mathbb{R}$ and $d \in \mathbb{N}$ parameters.
- Gaussian Kernel, also known as Radial Basis Function (RBF) kernel
$$K(u, v) = e^{-\frac{||u-v||^2}{\sigma^2}}$$
with $\sigma \in \mathbb{R}$ parameter.
- Sigmoid Kernel
$$K(u, v) = \tanh(\alpha \langle u, v \rangle + r)$$
for some (not every) $\alpha > 0$ and $r < 0$ parameters, where tanh is the hyperbolic tangent function.

The RBF is by far the most popular choice of kernel in Support Vector Machines. This is mainly because of its localized and finite response across the entire range of the real line; it also includes the polynomial Kernel as a limiting case. All these Kernels assume and need the data set features to be continuous.

3.3 Kernel density estimation (non parametric)

Kernel density estimation [2, 10, 11] allows density function $p(x)$ to be estimated using the dataset. One of the simplest approaches is the Parzen windows technique. For the univariate case is convenient to consider $\{x_1, \cdots, x_n\}$ i.i.d. sample of continuous random variable X for which the density $p(x)$ is unknown. As $p(x)$ is the derivative of the (cumulative) distribution function $P(x)$ (here $P_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{\{x_i \leq x\}}$ that means "proportion of points that are on the left of x") $\hat{p}(x)$ is defined as proportion of points in the interval $(x - h, x + h)$:

$$\hat{p}(x) = \frac{1}{2h}(P_n(x + h) - P_n(x - h))$$

or

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} \phi(z)$$

where $\phi(z) = \frac{1}{2}\mathbb{I}_{|z|\leq 1}$ is Parzen basis function and $z = \frac{x-x_i}{h}$ with h as a bandwidth parameter (to be estimated). The basis function must satisfy certain conditions of regularity [2, 10].

If the parameter h is too small, then the estimation of the density function leads to degeneration, on the the other had if it is too large, then the estimation is over-smoothed and it tends to uniform distribution.

The basis functions that have been used in this work are Gaussian and Epanechnikov [10] (their definition can be seen in the section of kernel extension).

Intuitively, $\hat{p}(x)$ with Gaussian basis function is the mean probability value of x following all possible ditributions, that is to say, centered on each known data point x_i with the same standard deviation. The Epanechnikov basis function is similar to Parzen window, with the difference that the distance to the points that lie in the window $(x - h, x + h)$ is weighted.

4 The missing value problem

Missing data arises in many statistical analyses nowadays. Absent information can be categorized as [14]:

- Missingness completely at random (MCAR)
- Missingness at random (MAR)
- Missingness that depends on unobserved predictors
- Missingness that depends on the missing value itself

MCAR happens when the probability of missingness is the same for all units. Throwing out cases with missing data does not bias inferences.

MAR is that the probability a variable is missing depends only on available information.

In the first case the missingness depends on information that has not been recorded and this information also predicts the missing values.

The last case is particularly difficult as the probability of missingness depends on the (potentially missing) variable itself.

In general, is very difficult to know whether data really are missing at random, or whether the missingness depends on unobserved predictors or the missing data themselves. In practice, as many predictors as possible are included.

Missing information is an old issue in statistical analysis **?**. For example, they are very common in Medicine and Engineering, where many variables come from on-line sensors or device measurements, or are simply too costly to be measured at the same rate as other variables (e.g., analytical tests). There are several causes for the absence of a value and they are so variate that we mention but a few:

- Technical limitations (e.g. sensors working only for given periods of time or sensor malfunctioning)
- Measures costly to perform in time or money or involving destructive methods (e.g., data from car crash tests)
- Measures not done by unknown number of reasons, or in invalid conditions, or simply lost during transmission or storage
- Senseless values related to other variables (e.g., number of pregnancies in male adults)
- Reluctance to supply the value (e.g., salaries, phone or credit card number, etc)

Missing information is difficult to handle, specially when the lost parts are of significant size. It can be either removed (the entire case) or “filled in” with the mean, median, nearest neighbour, or encoded by adding another input equal to one only if the value is absent and zero otherwise. Statistical approaches need to make parametric assumptions about or model the input distribution itself.

There are two basic ways of dealing with missing data:

1. Complete the object description in a hopefully optimal way
2. Extend the methods to be able to work with incomplete object descriptions

The possibility of simply discard the involved data can not be considered as a "method" and is also frustrating because of the lost effort in collecting the information. This can be done only if the number of missing values is very small or else they are heavily concentrated in some variables. In practice, it is not uncommon that missing values are distributed randomly (that is, according to an unknown distribution but independently of the observed values) and hence, if their quantity is not negligible, it is likely to affect a significant number of examples (in the worst scenario, just one missing value per example) so that discarding them all can not be afforded.

The single *Mean imputation* method seems the easiest way to impute, however this strategy can severely distort the distribution for this variable; for example, underestimating the standard deviation. Moreover, mean imputation distorts relationships between variables by "pulling" estimates of the correlation toward zero.

If the missing data are MAR, all simple techniques for handling missing data, complete and available case analyses, the indicator method and overall mean imputation, give biased results. When they are MCAR, these methods should be discarded too [16].

More complex method is the Multiple Imputation by Chained Equations [15], that consists of three steps:

- Imputation
- Analysis
- Pooling

The way of creating extended kernels for datasets with missing values has some important advantages:

- Any existent Kernel k can be extended to adapt to a dataset with missing values;
- No preprocessing of the missing values is needed; we create kernels by calculating directly the values of $k(x, \mathcal{X})$ and $k(\mathcal{X}, \mathcal{X})$ where x is a non-missing value and $\mathcal{X}$ represents a missing value, that is, without the need to estimate the value of $\mathcal{X}$ in every case. Moreover, there is no necessity of removing information because of the missing values; in other words, no information is lost.

- Missing values are allowed both in training and *test* examples (this is quite difficult with traditional imputation methods).

5 Kernel extension

In this section we extend the two more commonly used kernel functions for the SVM method; that are *RBF* (Radial Basis Function) and *Polynomial* kernels. Additionally, we need another kernel for each extension, that is the KDE kernel or, that is the same, the KDE basis function (for more details please check the section 3.3). In this work we have used two of them: *Gaussian* and *Epanechnikov.* Obviously, the first one assumes normality in the data, therefore we will see how the extension behaves over synthetic data set generated following the normal distribution (please check the section 7).

For clarity, the complete development is presented in 10, but only the initial form and the analytical result are exposed in this chapter. At the final of each subsection we reference the section with complete development.

The kernel extension for a single variable is proposed in the following form [2]:

$$\hat{k}(x,y) = \begin{cases} k(x,y) & \text{if } x,\ y \text{ are not missings} \\ \int_{-\infty}^{\infty} \hat{p}(x) k(x,y)\ dx & \text{if } x \text{ is missing} \\ \int_{-\infty}^{\infty} \hat{p}(y) k(x,y)\ dy & \text{if } y \text{ is missing} \\ \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{p}(x)\hat{p}(y) k(x,y)\ dxdy & \text{if } x,\ y \text{ are both missings} \end{cases}$$

The first case has no need for treatment, obviously. The second and the third ones are identical, so only will be seen the second case in detail. The fourth is an extension of the second and also will be seen in detail.

5.1 RBF (SVM kernel) - Gaussian (KDE kernel)

The RBF kernel has the form:

$$k(\boldsymbol{x},\boldsymbol{y}) = \exp\Big(-\frac{1}{2\sigma^2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\Big) = \exp\Big(-\frac{1}{2\sigma^2}\sum_{j=1}^{d}(x_j-y_j)^2\Big) = \prod_{j=1}^{d}\exp\Big(-\frac{1}{2\sigma^2}(x_j-y_j)^2\Big)$$

The gaussian KDE base function has the form:

$$\hat{p}(\boldsymbol{x}) = \prod_{j=1}^{d}\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{2h_j^2\pi}}\exp\Big(-\frac{1}{2h_j^2}\big(x_{ij}-x_j\big)^2\Big)$$

The extension for the case two has the form:

$$\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{d}\omega_j$$

where

$$\omega_j = \begin{cases} \frac{1}{(n-m_j)\sqrt{2h_j^2\pi}} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \delta & \text{if } x_j \text{ is missing} \\ \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j-y_j\big)^2\Big) & \text{if } x_j,\ y_j \text{ are not missings} \end{cases}$$

with $\delta = \exp\Big(-a\Big(\frac{\sigma_j^2 x_{ij}^2 + h_j^2 y_j^2}{h_j^2+\sigma_j^2}\Big) + ab^2\Big)\sqrt{\frac{\pi}{a}}$, $a = \frac{h_j^2+\sigma_j^2}{2h_j^2\sigma_j^2}$ and $b = \frac{h_j^2 y + \sigma_j^2 x_{ij}}{h_j^2+\sigma_j^2}$.

The extension for the case four has the form:

$$\hat{k}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{d} \omega_j$$

where

$$\omega_j = \begin{cases} \frac{1}{2(n-m_j)^2 h_j^2 \sqrt{a}\sqrt{a'}} \sum_{\substack{k=1\\ k\notin M_j}}^{n} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \delta & \text{if } x_j,\ y_j \text{ are missings} \\ \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j-y_j\big)^2\Big) & \text{if } x_j,\ y_j \text{ are not missings} \end{cases}$$

with $\delta = \exp\Big(a'b'^2 - \frac{y_{kj}^2 h_j^2 + y_{kj}^2\sigma_j^2 + x_{ij}^2 h_j^2}{2h_j^2(h_j^2+\sigma_j^2)}\Big)$, $a' = \frac{2\sigma_j^2 h_j^2 + \sigma_j^4}{2h_j^2\sigma_j^2(h_j^2+\sigma_j^2)}$ and $b' = \frac{y_{kj}\sigma_j^2 h_j^2 + y_{kj}\sigma_j^4 + \sigma_j^2 h_j^2 x_{ij}}{2\sigma_j^2 h_j^2 + \sigma_j^4}$. Please, notice that a and b have the same definition as in the case two.

For complete development, please check the section 10.1.

5.2 RBF (SVM kernel) - Epanechnikov (KDE kernel).

The kernel RBF is the same as in 5.1. The Epanechnikov KDE base function has the form:

$$\hat{p}(\boldsymbol{x}) = \prod_{j=1}^{d} \frac{3}{4h_j n} \sum_{i=1}^{n} \Big(1 - \Big(\frac{x_{ij}-x_j}{h_j}\Big)^2\Big)\mathbb{I}_{\left|\frac{x_{ij}-x_j}{h_j}\right|\leq 1}$$

The extension for the case two has the form:

$$\hat{k}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{d} \omega_j$$

where

$$\omega_j = \begin{cases} \frac{3}{4h_j(n-m_j)} \sum_{\substack{i=1\\ i\notin M_j}}^{n} I & \text{if } x_j \text{ is missing} \\ \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j-y_j\big)^2\Big) & \text{if } x_j,\ y_j \text{ are not missings} \end{cases}$$

with

$$I_{21} = \frac{x_{ij}^2}{h_j^2}\left[\frac{\sqrt{\pi}}{2\sqrt{a}}\mathrm{erf}\left((x-y)\sqrt{a}\right)\right]_c^b$$

$$I_{22} = \frac{2x_{ij}}{h_j^2}\left[-\frac{1}{2a}\exp(-a(x-y)^2) + \frac{y\sqrt{\pi}}{2\sqrt{a}}\mathrm{erf}\left((x-y)\sqrt{a}\right)\right]_c^b$$

$$\begin{aligned} I_{23} =& \frac{1}{h_j^2}\Bigg[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\mathrm{erf}\left((x-y)\sqrt{a}\right) - \frac{x}{2a}\exp(-a(x-y)^2) \\ &- \frac{y}{2a}\exp(-a(x-y)^2) + \frac{y^2\sqrt{\pi}}{2\sqrt{a}}\mathrm{erf}\left((x-y)\sqrt{a}\right)\Bigg]_c^b \end{aligned}$$

and $a = \frac{1}{2\sigma_j^2}$, $c = x_{ij} - h_j$ and $b = x_{ij} + h_j$, such that $I_2 = I_{21} - I_{22} + I_{23}$ and $I = I_1 - I_2$.

The extension for the case four has the form:

$$\hat{k}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{d} \omega_j$$

where

$$\omega_j = \begin{cases} \left(\frac{3}{4h_j(n-m_j)}\right)^2 \sum_{\substack{k=1 \\ k\notin M_j}}^{n} \sum_{\substack{i=1 \\ i\notin M_j}}^{n} I & \text{if } x_j,\, y_j \text{ are missings} \\ \exp\left(-\frac{1}{2\sigma_j^2}\left(x_j - y_j\right)^2\right) & \text{if } x_j,\, y_j \text{ are not missings} \end{cases}$$

with

$$\begin{aligned} I_1 =& \Bigg[\Big[\frac{h_j^2\sqrt{\pi} - y_{kj}^2\sqrt{\pi}}{2h_j^2\sqrt{a}}\int \mathrm{erf}\left((x-y)\sqrt{a}\right)\, dy \\ &- \frac{\sqrt{\pi}}{2h_j^2\sqrt{a}}\Big(\int y^2 erf((x-y)\sqrt{a})\, dy - 2y_{kj}\int y\mathrm{erf}\left((x-y)\sqrt{a}\right)\, dy\Big)\Big]_e^d\Bigg]_c^b \end{aligned}$$

$$
\begin{aligned}
I_2 = \Bigg[\Big[& \frac{x_{ij}^2\sqrt{\pi}}{2h_j^2\sqrt{a}} \int \operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy \\
& - \frac{2x_{ij}}{h_j^2}\Big(\frac{\sqrt{\pi}}{2\sqrt{a}} \int y\operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy - \frac{1}{2a}\int \exp(-a(x-y)^2)\ dy \Big) \\
& + \frac{1}{h_j^2}\Big(\frac{\sqrt{\pi}}{4\sqrt{a^3}} \int \operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy \\
& \qquad - \frac{x}{2a}\int \exp(-a(x-y)^2)\ dy \\
& \qquad - \frac{1}{2a}\int y\exp(-a(x-y)^2)\ dy \\
& \qquad + \frac{\sqrt{\pi}}{2\sqrt{a}} \int y^2\operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy \Big) \Big]_e^d \Bigg]_c^b
\end{aligned}
$$

$$
I_{31} = \frac{(x_{ij}y_{kj})^2}{h_j^4}\left[\Big[\frac{\sqrt{\pi}}{2\sqrt{a}} \int \operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy \Big]_e^d \right]_c^b
$$

$$
I_{32} = -\frac{2x_{ij}^2y_{kj}}{h_j^4}\left[\Big[\frac{\sqrt{\pi}}{2\sqrt{a}} \int y\operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy \Big]_e^d \right]_c^b
$$

$$
I_{33} = \frac{x_{ij}^2}{h_j^4}\left[\Big[\frac{\sqrt{\pi}}{2\sqrt{a}} \int y^2\operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy \Big]_e^d \right]_c^b
$$

$$
I_{34} = -\frac{2x_{ij}y_{kj}^2}{h_j^4}\left[\Big[\frac{\sigma_j\sqrt{\pi}}{\sqrt{2}} \int y\operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy - \sigma_j^2 \int \exp(-a(x-y)^2)\ dy \Big]_e^d \right]_c^b
$$

$$
I_{35} = \frac{4x_{ij}y_{kj}}{h_j^4}\left[\Big[\frac{\sigma_j\sqrt{\pi}}{\sqrt{2}} \int y^2\operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy - \sigma_j^2 \int y\exp(-a(x-y)^2)\ dy \Big]_e^d \right]_c^b
$$

$$
I_{36} = -\frac{2x_{ij}}{h_j^4}\left[\Big[\frac{\sigma_j\sqrt{\pi}}{\sqrt{2}} \int y^3\operatorname{erf}\left((x-y)\sqrt{a}\right)\ dy - \sigma_j^2 \int y^2\exp(-a(x-y)^2)\ dy \Big]_e^d \right]_c^b
$$

$$I_{37} = \frac{y_{kj}^2}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\int \mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy - \frac{x}{2a}\int \exp(-a(x-y)^2)\,dy$$
$$- \frac{1}{2a}\int y\exp(-a(x-y)^2)\,dy + \frac{\sqrt{\pi}}{2\sqrt{a}}\int y^2\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy\Big]_e^d\Bigg]_c^b$$

$$I_{38} = -\frac{2y_{kj}}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\int y\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy - \frac{x}{2a}\int y\exp(-a(x-y)^2)\,dy$$
$$- \frac{1}{2a}\int y^2\exp(-a(x-y)^2)\,dy + \frac{\sqrt{\pi}}{2\sqrt{a}}\int y^3\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy\Big]_e^d\Bigg]_c^b$$

$$I_{39} = \frac{1}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\int y^2\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy - \frac{x}{2a}\int y^2\exp(-a(x-y)^2)\,dy$$
$$- \frac{1}{2a}\int y^3\exp(-a(x-y)^2)\,dy + \frac{\sqrt{\pi}}{2\sqrt{a}}\int y^4\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy\Big]_e^d\Bigg]_c^b$$

and $a = \frac{1}{2\sigma_j^2}$, $e = y_{kj} - h_j$, $d = y_{kj} + h_j$, $c = x_{ij} - h_j$ and $b = x_{ij} + h_j$, such that $I_3 = I_{31} + I_{32} + I_{33} + I_{34} + I_{35} + I_{36} + I_{37} + I_{38} + I_{39}$ and $I = I_1 - I_2 + I_3$

For complete development, please check the section 10.2.

5.3 Polynomial (SVM kernel) - Gaussian (KDE kernel)

The Polynomial kernel has the form:

$$k(\boldsymbol{x},\boldsymbol{y}) = (\langle \boldsymbol{x},\boldsymbol{y}\rangle)^g = \prod_{j=1}^{g}\Big(\sum_{i_j=0}^{d} x_{i_j}y_{i_j}\Big) = \sum_{i_1=0}^{d}\cdot\sum_{i_2=0}^{d}\cdots\sum_{i_g=0}^{d}(x_{i_1}y_{i_1}\cdot x_{i_2}y_{i_2}\cdots x_{i_g}y_{i_g})$$

where g is the degree of the polynomial and $x_0 = y_0 = \sqrt{b}$ with $b \geq 0$ (the constant term of the polynomial).

The gaussian KDE base function is the same as in 5.1.

The extension is of the form:

$$\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \left[\sum_{i=0}^{d}\kappa_i\right]^g$$

where

$$\kappa_i = \begin{cases} I_{y_i} & \text{if } y_i \text{ is missings} \\ y_i \cdot \omega_i & \text{otherwise} \end{cases}$$

and

$$\omega_i = \begin{cases} I_{x_i} & \text{if } x_i \text{ is missings} \\ x_i & \text{otherwise} \end{cases}$$

with

$$I_{y_i} = \omega_i \frac{1}{(n - m_i)} \sum_{\substack{k=1 \\ k \notin M_i}}^{n} y_{ki}$$

$$I_{x_i} = \frac{1}{(n - m_i)} \sum_{\substack{k=1 \\ k \notin M_i}}^{n} x_{ki}$$

where m_i is the number of missings in the i-th variable and $M_i = \{i' : i' \in \{1..n\}, x_{i'i} \text{ is a missing}\}$.

For complete development, please check the section 10.3.

6 Implementation

Regarding the computational complexity, the solution for the SVM approach can be found in polynomial time. Thus, following a good programming practices the algorithm is straightforward. However, as it is described below, the choice of the programming language and different techniques were crucial in order to achieve good performance results, as some polynomial time algorithms may take days or even weeks to be computed.

All the kernel methods must compute a *Gram* matrix, that is the metrics matrix of the space of the data. Is a positive semi-definite symmetric matrix [13]. Its computation is the most time-consuming part when training an SVM model. It needed much effort to achieve good time performance.

The first goal was to implement the solution carefully, programming small scripts or functions ensuring this way their correctness by separate. That is to say, fulfilling the basic rule of modular programming.

The first attempt was to implement using symbolic calculus provided by Matlab. As this approach was too much time-consuming, several optimization have been done (please see the figure below):

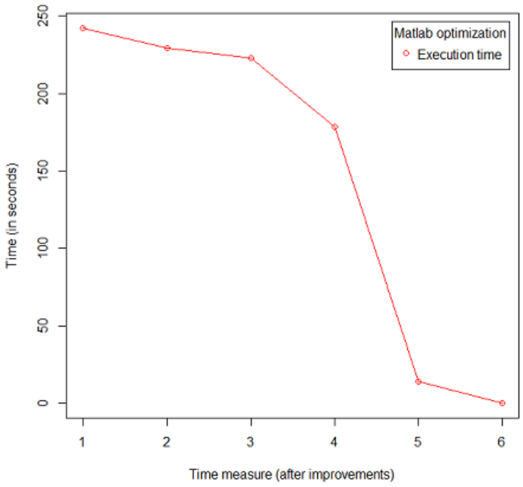


Figure 2: Execution time improvement in Matlab (50 × 2 data set).

where each cumulative optimization step (x label) means:

1. No optimization done
2. Partial loop vectorization
3. Complete loop vectorization
4. By reference mode of passing variables
5. Use of variables instead of symbolic expressions

6. Use of erf approximation [3] with three correct decimals

After measuring the time consumption of every function (extension case), the bottleneck turned out to be the 4-th case, as it was expected. In the latest optimization (erf approximation) the execution time had been reduced from 0.46s to 0.35s, and for big datasets it was not noticeable.

Finally, the decision was to implement this part in C++ and also to precalculate the 4-th case. These improvements have made the execution time to be very reasonable.

7 Experimentation

In this chapter we will perform an experimental study of the extended methods performance against other imputation methods. Thus we have kernel extensions that are (for more details please check the section 5):

1. RBF (SVM) - Gaussian (KDE)
2. RBF (SVM) - Epanechnikov (KDE)
3. Polynomial (SVM) - Gaussian (KDE)

And the imputation methods are:

1. MICE (Multivariate Imputation by Chained Equations) [15] - RBF (SVM)
2. Mean imputation - RBF (SVM)

Therefore, two data sets had been provided. One of them consisted of artificially generated data, while the second one was based on a real data [4].

The first and the third extensions are assuming normal distribution of the data, as they integrate the Gaussian KDE basis function.

MICE constructs a very complex models and its performance is expected to be good in any situation. By the contrary, Mean imputation is a very simple and straightforward way of univariate imputation.

The goal of this study is to see the effect of the missing value rate on the methods (mentioned above) performance and also to check the stability of the extended methods; that is to say, RBF - Gaussian extension should perform better than RBF - Epanechnikov on the synthetic data set (this will be discussed later). Moreover, the real-data data set allows us to compare the accuracy of the methods objectively.

7.1 Data sets

The artificial data set was generated from the gaussian distribution with the following parameters:

$$\boldsymbol{\mu}_1 = (0,0) \quad \boldsymbol{\mu}_2 = (3,3) \quad \Sigma_1 = \Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

where the subindex 1 is for the class $+1$ and the subindex 2 is for the class -1.

Training set size corresponds to $n = 200$. The test set is generated only once and it consists of 500 examples. The number of variables is two (without label) for 2-D representation purposes.

The missings are added randomly over the whole training and test data sets, thus the missing-data mechanism corresponds to MCAR. In order to perform more accurate and complete study, the training sets are sampled 10 times and the rate of missings ranges from 0% to 80% by steps of 10% for each of these sets.

The real data problem is a much studied dataset and represents a complex classification problem, in which a population of Pima Indian women living near Phoenix, Arizona, was tested for diabetes according to World Health Organization criteria.

The dataset contains 768 examples, 500 meaning negative conditions for diabetes (class 1) and 268 showing positive conditions of diabetes (class 2). Each example contains 8 attributes plus the class label.

In this data set, most of the variables show impossible zero values (e.g, the diastolic blood pressure), which are actually missing values. A more exhaustive analysis have been done, as can be seen in the table 1.

	Pregnancies	Pedigree	Age	Plasma	BMI	Blood	Skin	Serum	
392	1	1	1	1	1	1	1	1	0
1	1	1	1	0	1	1	1	1	1
140	1	1	1	1	1	1	1	0	1
1	1	1	1	1	0	1	1	1	1
4	1	1	1	0	1	1	1	0	2
2	1	1	1	1	1	0	1	0	2
192	1	1	1	1	1	1	0	0	2
1	1	1	1	1	0	1	1	0	2
26	1	1	1	1	1	0	0	0	3
2	1	1	1	1	0	1	0	0	3
7	1	1	1	1	0	0	0	0	4
	0	0	0	5	11	35	227	374	652

Table 1: This table shows how the missings are distributed over the data set.

It turns out that only 392 out of the 768 observations are unaffected by missing values. There are 51% of rows with missing values for which only Skin (Triceps skin fold thickness in *mm*) and Serum (2-Hour serum insulin in *mu U/ml*) are missing. In fact, this variables have the biggest proportion of missingnes through all the observations (30% and 49% respectively). The third less observed variable is the Blood (Diastolic blood pressure *mm Hg*) with 5%. Thus, the conclusion is that Skin and Serum are critical.

In 37% of rows with missings only Serum is not informed.

3 and 4 variables are not informed in 7% and 2% of rows with missings respectively.

The total proportion of missing values is $\approx 10.6\%$ against 48.9% of rows with missing values.

After this analysis it seems to be obvious that this difference is due to the high concentration of missings in the two critical variables, Skin and Serum.

7.2 Training

The training for extended kernel approach and imputation methods differs completely, as in the first case a *Gram* matrix is precomputed and in the second case it does `ksvm()` itself.

7.2.1 Training for extended kernels

For each kernel extension the corresponding training *Gram* matrix had been computed, that is, only over the training examples. There are three parameters to take into account, that are bandwidth KDE parameter, σ RBF kernel parameter (where it appears as $\gamma = \frac{1}{2\sigma^2}$) and polynomial degree and offset parameters for Polynomial kernel.

The first one is inherent to kernel extension and is estimated with `density()` [7], giving as parameters bandwidth smoothing and kernel or window basis function (Gaussian and Epanechnikov have been proposed in this work). The rest of parameters are so called hyperparameters and, usually, are tuned by means of k-fold cross-validation technique, that is explained below. In this work and for computational purposes the σ RBF kernel parameter had been estimated over training data set with `sigest()` [9] from kernlab package and the polynomial degree parameter was set by k-fold cross-validation, the offset was set to 1 by default.

For the regularization parameter $C - cost$ k-fold cross-validation method had been applied.

7.2.2 Training for imputed data sets

In the case of imputed data is slightly different. The model is trained with RBF predefined kernel. For this purpose, the imputation must be performed against both sets (training and test), but separately.

For the case of MICE imputation, the parameters are the number of imputations for each missing (5 by default) and predictor matrix, that describes the relation between the variables, thus only some of the variables might be used for the imputation of another variable. For example, the diagonal is zero, as no variable predicts itself.

For the case of Mean imputation, there are no parameters.

The σ parameter had been estimated as in the extension case, but over the whole training data matrix. The regularization parameter $C-cost$ had been tuned through k-fold cross-validation.

6.2.3 K-fold cross-validation technique

This technique is widely used for the hyperparameter estimation. The model is sequentially trained over each $k-1$ training data set folds and then tested against the fold that had not been used for training. This is performed for each $C - cost$ value. In fact, this technique is used to estimate any parameter that needs to be tuned. Finally, the best parameter is selected comparing the (mean) cross-validation error.

7.3 Test

The test corresponds to the weighted summation of the evaluation of the kernel function for test examples against the support vectors found in the training step.

7.4 Statistical study of the error rates

Depending on the kind of the data set, I applied different types of study:

1. Artificial data set: boxplot for each method and number of examples.
2. Real-data data set: confidence interval and paired t-test [6].

7.4.1 Confidence interval definition

The goodness of the prediction is measured with a variable that measures a number of successfull predictions and follows the binomial distribution (n, p). By the CLT, as n is big enough, the distribution of the sample mean will be approximately normal. It can also be seen as a summation of n random independent variables with Bernoulli distribution. Then, the proportion is the mean of this summation and it corresponds to the sample mean that follows $N(\mu = p, \sigma = \sqrt{\frac{p(1-p)}{n}})$. I set $p = \hat{p}$, where $\hat{p}$ is the measured error rate.

The confidence interval at 95% is defined as follows

$$p \in \left[\hat{p} - z_{\alpha/2} \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z_{\alpha/2} \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right]$$

with the significance level $\alpha = 0.05$.

7.4.2 Paired t-test definition

A paired data sample is a set of observations of two variables (here methods) sampled over the same individuals, so that there are two observations for each individual. In order to perform the paired t-test, first of all I have converted two samples in one sample as a result of the difference. The difference must be done as:

$$\text{difference} = \text{extended kernel error rate} - \text{imputed error rate}$$

Then, the paired t-test is defined as follows (n is big enough or normality assumption):

$$H_0 : \mu = 0 \quad \text{against} \quad H_1 : \mu > 0$$

the test statistic is

$$\frac{\sqrt{n}(\hat{x} - \mu_0)}{s}$$

where $\hat{x}$ is the mean of the differences and s is the sample standard deviation. To refuse the null hypothesis the inequality must hold:

$$1 - P\left(t_{n-1} \leq \frac{\sqrt{n}(\hat{x} - \mu_0)}{s}\right) < \alpha$$

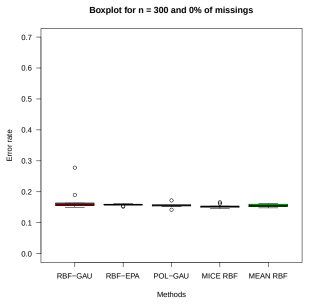
with $\alpha = 0.05$ as a significance level.

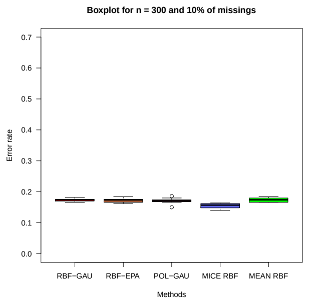
8 Results

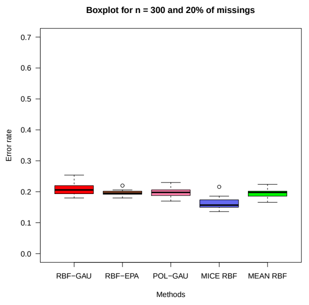
In this section we present several tables and plots in order to perform a reliable comparative between methods analyzed in this work. Further on, the results will be discussed in the sub section 8.3.

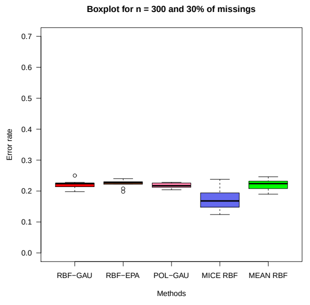
8.1 Results for the synthetic-data data set

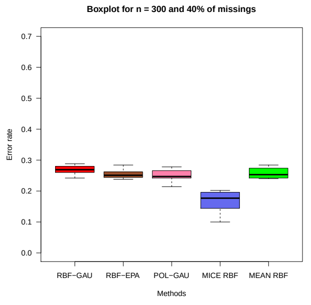
In order to compare the performance of every method, we have constructed a boxplot for each missing rate and method; that is, ranging from 0% to 80%. Additionally, we have constructed a plot which shows the effect of the missing rates on every method. In this way we will be able to check the stability of the methods and also their consistency; that is to say, not oscilating behaviour or at least not too much.

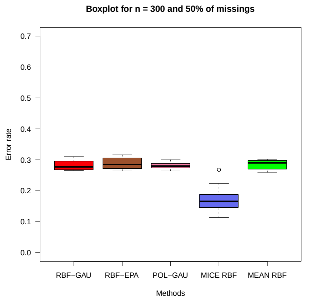












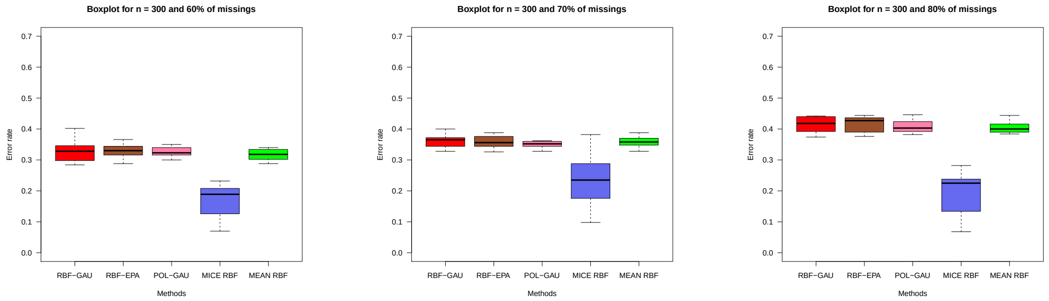


Figure 3: Boxplots to measure a general performance between methods for each missing rate.

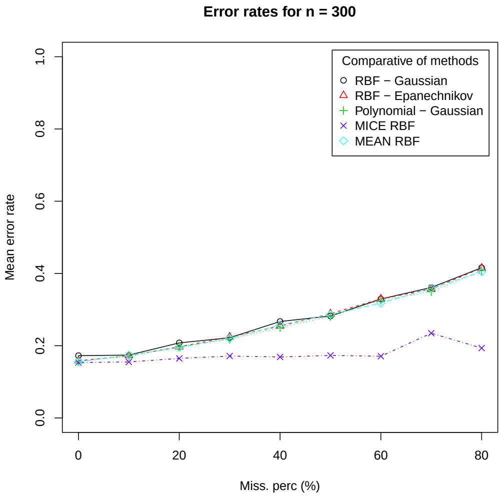


Figure 4: Comparative of different methods over sinthetic data varying the missing rate.

8.2 Results for the real-data data set

Here we present two measures for the analysis of the performance of the methods on a real-data data set. The first table shows the confidence interval with 5% of confidence level. In order to build it we had needed at least 30 different samples (here training sets).

Method	CI at 95%
RBF (SVM) - Gaussian (KDE)	$[0.2546, 0.2726]$
RBF (SVM) - Epanechnikov (KDE)	$[0.2542, 0.2722]$
Polynomial (SVM) - Gaussian (KDE)	$[0.2713, 0.2896]$
MICE - RBF (SVM)	$[0.2471, 0.2649]$
Mean - RBF (SVM)	$[0.2376, 0.2552]$

Table 2: Confidence intervals for real-data data set

In the following table is shown the result of the paired t-test. Performing this test we are expecting to check if the difference between the extension methods error rates and the imputation methods error rates is significant. We had chosen the significance level of 5% to be careful accepting the null hypothesis when in fact there is a difference.

	MICE - RBF (SVM)	Mean - RBF (SVM)
RBF (SVM) - Gaussian (KDE)	$0.4535 \cdot 10^{-1}$	$0.7851 \cdot 10^{-4}$
RBF (SVM) - Epanechnikov (KDE)	$0.787 \cdot 10^{-1}$	$0.3282 \cdot 10^{-3}$
Polynomial (SVM)- Gaussian (KDE)	$0.4187 \cdot 10^{-4}$	$0.3749 \cdot 10^{-8}$

Table 3: p-value of paired t-test with $\alpha = 0.05$ (extended kernels against imputation methods).

8.3 Discussion

With respect to a synthetic-data problem, the three extension methods perform similarly to the Mean imputation method. Their interquartile range is very compressed independently of the missing rate. But, in this case, the MICE - RBF (SVM) method seems to be the most robust one. As can be seen in the figure 4, it maintains the accuracy even incrementing the missing rate. However, its interquartile range is much wider and for 30% and 70% of missings its maximum error rate can be as high as other methods.

Regarding the real-data problem, the method Polynomial (SVM) - Gaussian (KDE) had the worst behavior, as it shows its confidence interval in the table 2. It turns out that this method is similar to the Mean imputation approach, but using Polynomial kernel function instead of RBF kernel, as it is done in this work. The results differ significatively between these two methods, as can be seen in the table 3.

The other two extensions behave similarly (very similar confidence interval) and also we can not afirm that there is a significant difference between these two methods and MICE - RBF (SVM) method, as show their p-values in the table 3.

It appears to be that the Mean - RBF (SVM) method works better in the real-data problem than even MICE - RBF (SVM). On the other hand, in the synthetic-data problem the best performance shows MICE imputation method, as it was expected.

9 Conclusions and future work

This work have consisted mostly in theoretical development under the guidance of my teacher. The approach that has been presented is considered novel in the literature and there is much work to do.

The main goal of practical part was to make a feasible solution from the computational point of view. There is much room for improvement and optimization.

Moreover, as have been seen in section 8.3, all three kernel extensions are competitive with standard imputation methods. In the synthetic-data problem they behave similarly to Mean - RBF (SVM) method and in the real-data problem, statistical analyses show that they are near to the MICE - RBF (SVM) method.

The advantages that present kernel extension methods are the following:

- They do not modify the data
- They are completely deterministic
- They allow to predict data with missings (even one observation)
- They are well formed theoretically

There can be done several improvements in the future:

- Normalize the kernels
- Optimize σ parameter
- Improve the efficiency
- Test with more problems

10 Annex

10.1 RBF (SVM kernel) - Gaussian (KDE kernel)

Case 2:

$$\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{d} \int_{-\infty}^{\infty} \frac{1}{n-m_j} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \frac{1}{\sqrt{2h_j^2\pi}} \exp\Big(-\frac{1}{2h_j^2}\big(x_{ij}-x_j\big)^2\Big) \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j-y_j\big)^2\Big)\, dx_j$$

where m_j is the number of missings in the j-th variable and $M_j = \{i' : i' \in \{1..n\},\ x_{i'j}$ is a missing$\}$. All d integrals are identical, thus only one of them will be solved.

$$\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \int_{-\infty}^{\infty} \frac{1}{n-m_j} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \frac{1}{\sqrt{2h_j^2\pi}} \exp\Big(-\frac{1}{2h_j^2}\big(x_{ij}-x\big)^2\Big) \exp\Big(-\frac{1}{2\sigma_j^2}\big(x-y\big)^2\Big)\, dx$$

with $x = x_j$ and $y = y_j$ (for simplicity).

By the linearity of the antidifferentiation:

$$\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{(n-m_j)\sqrt{2h_j^2\pi}} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \int_{-\infty}^{\infty} \exp\Big(-\frac{1}{2h_j^2}\big(x_{ij}-x\big)^2\Big) \exp\Big(-\frac{1}{2\sigma_j^2}\big(x-y\big)^2\Big)\, dx$$

that performing simple algebraic operations conduces to

$$\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{(n-m_j)\sqrt{2h_j^2\pi}} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \exp\left(-a\left(\frac{\sigma_j^2 x_{ij}^2 + h_j^2 y^2}{h_j^2+\sigma_j^2}\right)\right) \int_{-\infty}^{\infty} \exp\Big(-a\big(x^2-2bx\big)\Big)\, dx$$

where $a = \frac{h_j^2+\sigma_j^2}{2h_j^2\sigma_j^2}$ and $b = \frac{h_j^2 y+\sigma_j^2 x_{ij}}{h_j^2+\sigma_j^2}$.

$$
\begin{aligned}
I =& \int_{-\infty}^{\infty} \exp\Big(-a\big(x^2-2bx\big)\Big)\ dx\\
=& \exp\Big(ab^2\Big)\int_{-\infty}^{\infty} \exp\Big(-a\big(x-b\big)^2\Big)\ dx\\
=& \exp\Big(ab^2\Big)\int_{-\infty}^{\infty} \exp\Big(-at^2\Big)\ dt\\
=& \exp\Big(ab^2\Big)\sqrt{\frac{\pi}{a}}
\end{aligned}
$$

where $t = x - b$ and $dt = \frac{\partial}{\partial x}(x-b) = dx$. For more details, please check 10.4.

Finally, the kernel extension expression is:

$$
\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{d} \omega_j
$$

where

$$
\omega_j = \begin{cases} \frac{1}{(n-m_j)\sqrt{2h_j^2\pi}} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \delta & \text{if } x_j \text{ is missing}\\ \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j-y_j\big)^2\Big) & \text{if } x_j,\ y_j \text{ are not missings} \end{cases}
$$

with $\delta = \exp\Big(-a\Big(\frac{\sigma_j^2 x_{ij}^2 + h_j^2 y_j^2}{h_j^2+\sigma_j^2}\Big) + ab^2\Big)\sqrt{\frac{\pi}{a}}$.

Case 4:

$$
\begin{aligned}
\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{d} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} & \frac{1}{(n-m_j)} \sum_{\substack{k=1\\ k\notin M_j}}^{n} \frac{1}{\sqrt{2h_j^2\pi}} \exp\Big(-\frac{1}{2h_j^2}\big(y_{kj}-y_j\big)^2\Big)\\
& \frac{1}{(n-m_j)} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \frac{1}{\sqrt{2h_j^2\pi}} \exp\Big(-\frac{1}{2h_j^2}\big(x_{ij}-x_j\big)^2\Big)\\
& \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j-y_j\big)^2\Big)\ dx_j dy_j
\end{aligned}
$$

Again, is considered only one integral. Performing substitution of the result obtained in case 2:

$$
\begin{aligned}
\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \int_{-\infty}^{\infty} & \frac{1}{(n-m_j)\sqrt{2h_j^2\pi}} \sum_{\substack{k=1\\ k\notin M_j}}^{n} \exp\Big(-\frac{1}{2h_j^2}\big(y_{kj}-y\big)^2\Big)\cdot \\
& \cdot \frac{1}{(n-m_j)\sqrt{2h_j^2\pi}} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \exp\left(-a\left(\frac{\sigma_j^2 x_{ij}^2 + h_j^2 y^2}{h_j^2+\sigma_j^2}\right)\right) \exp(ab^2)\sqrt{\frac{\pi}{a}}\ dy \\
= & \frac{\sqrt{\pi}}{\Big((n-m_j)\sqrt{2h_j^2\pi}\Big)^2\sqrt{a}} \sum_{\substack{k=1\\ k\notin M_j}}^{n} \sum_{\substack{i=1\\ i\notin M_j}}^{n} \int_{-\infty}^{\infty} \exp\Big(-\frac{1}{2h_j^2}\big(y_{kj}-y\big)^2\Big)\cdot \\
& \qquad\qquad \cdot \exp\left(-a\left(\frac{\sigma_j^2 x_{ij}^2 + h_j^2 y^2}{h_j^2+\sigma_j^2}\right)\right) \\
& \qquad\qquad \cdot \exp(ab^2)\ dy
\end{aligned}
$$

I take

$$
I = \int_{-\infty}^{\infty} \exp\Big(-\frac{1}{2h_j^2}\big(y_{kj}-y\big)^2 - a\left(\frac{\sigma_j^2 x_{ij}^2 + h_j^2 y^2}{h_j^2+\sigma_j^2}\right) + ab^2\Big)\ dy
$$

and applying simple algebraic operations

$$
I = \exp\Big(-\frac{y_{kj}^2 h_j^2 + y_{kj}^2\sigma_j^2 + x_{ij}^2 h_j^2}{2h_j^2(h_j^2+\sigma_j^2)}\Big) \int_{-\infty}^{\infty} \exp\big(-a'(y^2+2b'y)\big)\ dy
$$

where $a' = \frac{2\sigma_j^2 h_j^2 + \sigma_j^4}{2h_j^2\sigma_j^2(h_j^2+\sigma_j^2)}$ and $b' = \frac{y_{kj}\sigma_j^2 h_j^2 + y_{kj}\sigma_j^4 + \sigma_j^2 h_j^2 x_{ij}}{2\sigma_j^2 h_j^2 + \sigma_j^4}$.

$$
\begin{aligned}
I = & \exp(a'b'^2)\exp\Big(-\frac{y_{kj}^2 h_j^2 + y_{kj}^2\sigma_j^2 + x_{ij}^2 h_j^2}{2h_j^2(h_j^2+\sigma_j^2)}\Big) \int_{-\infty}^{\infty} \exp\big(-a'(y+b)^2\big)\ dy \\
= & \exp(a'b'^2)\exp\Big(-\frac{y_{kj}^2 h_j^2 + y_{kj}^2\sigma_j^2 + x_{ij}^2 h_j^2}{2h_j^2(h_j^2+\sigma_j^2)}\Big) \int_{-\infty}^{\infty} \exp\big(-a't^2\big)\ dt \\
= & \exp(a'b'^2)\exp\Big(-\frac{y_{kj}^2 h_j^2 + y_{kj}^2\sigma_j^2 + x_{ij}^2 h_j^2}{2h_j^2(h_j^2+\sigma_j^2)}\Big) \sqrt{\frac{\pi}{a'}}
\end{aligned}
$$

where $t = y + b'$ and $dt = \frac{\partial}{\partial y}(y+b') = dy$. For more details, please check 10.4. Finally, the kernel extension expression is:

$$
\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{d} \omega_j
$$

where

$$\omega_j = \begin{cases} \frac{1}{2(n-m_j)^2 h_j^2 \sqrt{a}\sqrt{a'}} \sum_{\substack{k=1 \\ k\notin M_j}}^n \sum_{\substack{i=1 \\ i\notin M_j}}^n \delta & \text{if } x_j,\, y_j \text{ are missings} \\ \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j - y_j\big)^2\Big) & \text{if } x_j,\, y_j \text{ are not missings} \end{cases}$$

with $\delta = \exp\Big(a'b'^2 - \frac{y_{kj}^2 h_j^2 + y_{kj}^2\sigma_j^2 + x_{ij}^2 h_j^2}{2h_j^2(h_j^2+\sigma_j^2)}\Big)$.

10.2 RBF (SVM kernel) - Epanechnikov (KDE kernel)

Case 2:

Is considered only one integral as in 10.1.

$$\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \int_{-\infty}^{\infty} \frac{3}{4h_j(n-m_j)} \sum_{\substack{i=1 \\ i\notin M_j}}^n \Big(1 - \Big(\frac{x_{ij}-x}{h_j}\Big)^2\Big)\mathbb{I}_{\left|\frac{x_{ij}-x}{h_j}\right|\leq 1} \cdot \exp\Big(-a\big(x-y\big)^2\Big)\, dx$$

where $x = x_j$, $y = y_j$ and $a = \frac{1}{2\sigma_j^2}$. Manipulating the terms is obtained the following expression:

$$\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \frac{3}{4h_j(n-m_j)} \sum_{\substack{i=1 \\ i\notin M_j}}^n \int_c^b \Big(1 - \Big(\frac{x_{ij}-x}{h_j}\Big)^2\Big) \cdot \exp\Big(-a\big(x-y\big)^2\Big)\, dx$$

where $c = x_{ij} - h_j$ and $b = x_{ij} + h_j$.

I take the following integrals

$$\begin{aligned} I &= \int_c^b \Big(1 - \Big(\frac{x_{ij}-x}{h_j}\Big)^2\Big) \cdot \exp\Big(-a\big(x-y\big)^2\Big)\, dx \\ &= \int_c^b \exp\Big(-a\big(x-y\big)^2\Big)\, dx - \int_c^b \Big(\frac{x_{ij}-x}{h_j}\Big)^2 \cdot \exp\Big(-a\big(x-y\big)^2\Big)\, dx \end{aligned}$$

,

$$\begin{aligned} I_1 &= \int_c^b \exp\Big(-a\big(x-y\big)^2\Big)\, dx \\ &= \frac{\sqrt{\pi}}{2\sqrt{a}}\Big[\operatorname{erf}\left((x-y)\sqrt{a}\right)\Big]_c^b \end{aligned}$$

$$I_2 = \int_c^b \Big(\frac{x_{ij} - x}{h_j}\Big)^2 \cdot \exp\Big(- a\big(x-y\big)^2\Big)\ dx$$

and

$$\begin{aligned}
I_{21} =& \frac{x_{ij}^2}{h_j^2}\int_c^b \exp\Big(- a\big(x-y\big)^2\Big)\ dx \\
=& \frac{x_{ij}^2}{h_j^2}\left[\frac{\sqrt{\pi}}{2\sqrt{a}}\text{erf}\left((x-y)\sqrt{a}\right)\right]_c^b
\end{aligned}$$

$$\begin{aligned}
I_{22} =& \frac{2x_{ij}}{h_j^2}\int_c^b x\cdot\exp\Big(- a\big(x-y\big)^2\Big)\ dx \\
=& \frac{2x_{ij}}{h_j^2}\left[-\frac{1}{2a}\exp(-a(x-y)^2) + \frac{y\sqrt{\pi}}{2\sqrt{a}}\text{erf}\left((x-y)\sqrt{a}\right)\right]_c^b
\end{aligned}$$

$$\begin{aligned}
I_{23} =& \frac{1}{h_j^2}\int_c^b x^2\cdot\exp\Big(- a\big(x-y\big)^2\Big)\ dx \\
=& \frac{1}{h_j^2}\left[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\text{erf}\left((x-y)\sqrt{a}\right) - \frac{x}{2a}\exp(-a(x-y)^2)\right. \\
& \left. - \frac{y}{2a}\exp(-a(x-y)^2) + \frac{y^2\sqrt{\pi}}{2\sqrt{a}}\text{erf}\left((x-y)\sqrt{a}\right)\right]_c^b
\end{aligned}$$

such that $I_2 = I_{21} - I_{22} + I_{23}$ and $I = I_1 - I_2$. For more details, please check 10.4. Finally, the kernel extension expression is:

$$\hat{k}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{d} \omega_j$$

where

$$\omega_j = \begin{cases} \frac{3}{4h_j(n-m_j)}\sum_{\substack{i=1 \\ i\notin M_j}}^n I & \text{if } x_j \text{ is missing} \\ \exp\Big(- \frac{1}{2\sigma_j^2}\big(x_j - y_j\big)^2\Big) & \text{if } x_j,\, y_j \text{ are not missings} \end{cases}$$

Case 4:

$$\begin{aligned}\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} & \frac{3}{4h_j(n-m_j)} \sum_{\substack{k=1\\k\notin M_j}}^{n} \Big(1-\Big(\frac{y_{kj}-y}{h_j}\Big)^2\Big)\mathbb{I}_{|\frac{y_{kj}-y}{h_j}|\le 1} \\ & \cdot \frac{3}{4h_j(n-m_j)} \sum_{\substack{i=1\\i\notin M_j}}^{n} \Big(1-\Big(\frac{x_{ij}-x}{h_j}\Big)^2\Big)\mathbb{I}_{|\frac{x_{ij}-x}{h_j}|\le 1} \\ & \cdot \exp\Big(-a\big(x-y\big)^2\Big)\ dxdy\end{aligned}$$

where $a = \frac{1}{2\sigma_j^2}$, $x = x_j$ and $y = y_j$. Manipulating the expression above

$$\hat{k}_j(\boldsymbol{x},\boldsymbol{y}) = \Big(\frac{3}{4h_j(n-m_j)}\Big)^2 \sum_{\substack{k=1\\k\notin M_j}}^{n}\sum_{\substack{i=1\\i\notin M_j}}^{n} \int_e^d\int_c^b \delta_{x,y}\ dxdy$$

where $\delta_{x,y} = \Big(\Big(1-\big(\frac{y_{kj}-y}{h_j}\big)^2\Big) - \big(\frac{x_{ij}-x}{h_j}\big)^2\Big(1-\big(\frac{y_{kj}-y}{h_j}\big)^2\Big)\Big)\exp\Big(-a\big(x-y\big)^2\Big)$, $e = y_{kj} - h_j$, $d = y_{kj} + h_j$, $c = x_{ij} - h_j$ and $b = x_{ij} + h_j$.

I take the following integrals

$$\begin{aligned}I_1 =& \int_e^d\int_c^b \Big(1-\Big(\frac{y_{kj}-y}{h_j}\Big)^2\Big)\exp\big(-a\big(x-y\big)^2\big)\ dxdy \\ =& \int_e^d \Big(1-\Big(\frac{y_{kj}-y}{h_j}\Big)^2\Big)\frac{\sqrt{\pi}}{2\sqrt{a}}\big[\text{erf}\,((x-y)\sqrt{a})\big]_c^b\ dy \\ =& \Bigg[\Big[\frac{h_j^2\sqrt{\pi}-y_{kj}^2\sqrt{\pi}}{2h_j^2\sqrt{a}}\int \text{erf}\,((x-y)\sqrt{a})\ dy \\ & -\frac{\sqrt{\pi}}{2h_j^2\sqrt{a}}\Big(\int y^2 erf((x-y)\sqrt{a})\ dy - 2y_{kj}\int y\text{erf}\,((x-y)\sqrt{a})\ dy\Big)\Big]_e^d\Bigg]_c^b\end{aligned}$$

$$
\begin{aligned}
I_2 =& \int_e^d \int_c^b \Big(\frac{x_{ij}-x}{h_j}\Big)^2 \exp\big(-a\big(x-y\big)^2\big)\;dxdy \\
=& \Bigg[\Big[\frac{x_{ij}^2\sqrt{\pi}}{2h_j^2\sqrt{a}}\int \mathrm{erf}\,((x-y)\sqrt{a})\;dy \\
&- \frac{2x_{ij}}{h_j^2}\Big(\frac{\sqrt{\pi}}{2\sqrt{a}}\int y\mathrm{erf}\,((x-y)\sqrt{a})\;dy - \frac{1}{2a}\int \exp(-a(x-y)^2)\;dy\Big) \\
&+ \frac{1}{h_j^2}\Big(\frac{\sqrt{\pi}}{4\sqrt{a^3}}\int \mathrm{erf}\,((x-y)\sqrt{a})\;dy \\
&\qquad - \frac{x}{2a}\int \exp(-a(x-y)^2)\;dy \\
&\qquad - \frac{1}{2a}\int y\exp(-a(x-y)^2)\;dy \\
&\qquad + \frac{\sqrt{\pi}}{2\sqrt{a}}\int y^2\mathrm{erf}\,((x-y)\sqrt{a})\;dy\Big)\Big]_e^d\Bigg]_c^b
\end{aligned}
$$

and

$$I_3 = \int_e^d \int_c^b \Big(\frac{(x_{ij}-x)(y_{kj}-y)}{h_j^2}\Big)^2 \exp\big(-a\big(x-y\big)^2\big)\;dxdy$$

that can be decomposed into the following integrals

$$
\begin{aligned}
I_{31} =& \frac{(x_{ij}y_{kj})^2}{h_j^4}\int_e^d \int_c^b \exp\big(-a\big(x-y\big)^2\big)\;dxdy \\
=& \frac{(x_{ij}y_{kj})^2}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{2\sqrt{a}}\int \mathrm{erf}\,((x-y)\sqrt{a})\;dy\Big]_e^d\Bigg]_c^b
\end{aligned}
$$

$$
\begin{aligned}
I_{32} =& -\frac{2x_{ij}^2y_{kj}}{h_j^4}\int_e^d \int_c^b y\exp\big(-a\big(x-y\big)^2\big)\;dxdy \\
=& -\frac{2x_{ij}^2y_{kj}}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{2\sqrt{a}}\int y\mathrm{erf}\,((x-y)\sqrt{a})\;dy\Big]_e^d\Bigg]_c^b
\end{aligned}
$$

$$
\begin{aligned}
I_{33} =& \frac{x_{ij}^2}{h_j^4}\int_e^d \int_c^b y^2\exp\big(-a\big(x-y\big)^2\big)\;dxdy \\
=& \frac{x_{ij}^2}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{2\sqrt{a}}\int y^2\mathrm{erf}\,((x-y)\sqrt{a})\;dy\Big]_e^d\Bigg]_c^b
\end{aligned}
$$

$$\begin{aligned} I_{34} =& -\frac{2x_{ij}y_{kj}^2}{h_j^4}\int_e^d\int_c^b x\exp\big(-a\big(x-y\big)^2\big)\ dxdy \\ =& -\frac{2x_{ij}y_{kj}^2}{h_j^4}\Bigg[\Big[\frac{\sigma_j\sqrt{\pi}}{\sqrt{2}}\int y\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy-\sigma_j^2\int\exp(-a(x-y)^2)\ dy\Big]_e^d\Bigg]_c^b \end{aligned}$$

$$\begin{aligned} I_{35} =& \frac{4x_{ij}y_{kj}}{h_j^4}\int_e^d\int_c^b yx\exp\big(-a\big(x-y\big)^2\big)\ dxdy \\ =& \frac{4x_{ij}y_{kj}}{h_j^4}\Bigg[\Big[\frac{\sigma_j\sqrt{\pi}}{\sqrt{2}}\int y^2\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy-\sigma_j^2\int y\exp(-a(x-y)^2)\ dy\Big]_e^d\Bigg]_c^b \end{aligned}$$

$$\begin{aligned} I_{36} =& -\frac{2x_{ij}}{h_j^4}\int_e^d\int_c^b y^2x\exp\big(-a\big(x-y\big)^2\big)\ dxdy \\ =& -\frac{2x_{ij}}{h_j^4}\Bigg[\Big[\frac{\sigma_j\sqrt{\pi}}{\sqrt{2}}\int y^3\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy-\sigma_j^2\int y^2\exp(-a(x-y)^2)\ dy\Big]_e^d\Bigg]_c^b \end{aligned}$$

$$\begin{aligned} I_{37} =& \frac{y_{kj}^2}{h_j^4}\int_e^d\int_c^b x^2\exp\big(-a\big(x-y\big)^2\big)\ dxdy \\ =& \frac{y_{kj}^2}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\int\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy-\frac{x}{2a}\int\exp(-a(x-y)^2)\ dy \\ &\qquad -\frac{1}{2a}\int y\exp(-a(x-y)^2)\ dy+\frac{\sqrt{\pi}}{2\sqrt{a}}\int y^2\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy\Big]_e^d\Bigg]_c^b \end{aligned}$$

$$\begin{aligned} I_{38} =& -\frac{2y_{kj}}{h_j^4}\int_e^d\int_c^b yx^2\exp\big(-a\big(x-y\big)^2\big)\ dxdy \\ =& -\frac{2y_{kj}}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\int y\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy-\frac{x}{2a}\int y\exp(-a(x-y)^2)\ dy \\ &\qquad -\frac{1}{2a}\int y^2\exp(-a(x-y)^2)\ dy+\frac{\sqrt{\pi}}{2\sqrt{a}}\int y^3\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy\Big]_e^d\Bigg]_c^b \end{aligned}$$

$$
\begin{aligned}
I_{39} =&\frac{1}{h_j^4}\int_e^d\int_c^b y^2x^2\exp\big(-a\big(x-y\big)^2\big)\ dxdy\\
=&\frac{1}{h_j^4}\Bigg[\Big[\frac{\sqrt{\pi}}{4\sqrt{a^3}}\int y^2\text{erf}\left((x-y)\sqrt{a}\right)\ dy-\frac{x}{2a}\int y^2\exp(-a(x-y)^2)\ dy\\
&\qquad -\frac{1}{2a}\int y^3\exp(-a(x-y)^2)\ dy+\frac{\sqrt{\pi}}{2\sqrt{a}}\int y^4\text{erf}\left((x-y)\sqrt{a}\right)\ dy\Big]_e^d\Bigg]_c^b
\end{aligned}
$$

such that $I_3 = I_{31} + I_{32} + I_{33} + I_{34} + I_{35} + I_{36} + I_{37} + I_{38} + I_{39}$ and $I = I_1 - I_2 + I_3$.

Finally, the kernel extension expression is:

$$
\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{d}\omega_j
$$

where

$$
\omega_j = \begin{cases} \left(\frac{3}{4h_j(n-m_j)}\right)^2\sum_{\substack{k=1\\k\notin M_j}}^n\sum_{\substack{i=1\\i\notin M_j}}^n I & \text{if } x_j,\ y_j \text{ are missings}\\ \exp\Big(-\frac{1}{2\sigma_j^2}\big(x_j-y_j\big)^2\Big) & \text{if } x_j,\ y_j \text{ are not missings}\end{cases}
$$

10.3 Polynomial (SVM kernel) - Gaussian (KDE kernel)

For construction of the extension I will assume that both variables are completely missing, except the constant term, and then two more variables will be introduced in order to manage all the possible cases.

By the linearity of the antidifferentiation, the extension has the following form:

$$
\begin{aligned}
\hat{k}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i_1=0}^{d}\cdot\sum_{i_2=0}^{d}\cdots\sum_{i_g=0}^{d}&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x_{i_1}y_{i_1}\hat{p}(x_{i_1})\hat{p}(y_{i_1})\ dx_{i_1}\ dy_{i_1}\cdot\\
&\cdot\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x_{i_2}y_{i_2}\hat{p}(x_{i_2})\hat{p}(y_{i_2})\ dx_{i_2}\ dy_{i_2}\cdots\\
&\cdot\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x_{i_g}y_{i_g}\hat{p}(x_{i_g})\hat{p}(y_{i_g})\ dx_{i_g}\ dy_{i_g}
\end{aligned}
$$

Grouping the terms the expression is (here the first auxiliar variable is introduced)

$$
\begin{aligned}
\hat{k}(\boldsymbol{x},\boldsymbol{y}) = & \sum_{i_1=0}^{d} \int_{-\infty}^{\infty} y_{i_1}\hat{p}(y_{i_1}) \int_{-\infty}^{\infty} x_{i_1}\hat{p}(x_{i_1})\ dx_{i_1}\ dy_{i_1} \cdot \\
& \sum_{i_2=0}^{d} \int_{-\infty}^{\infty} y_{i_2}\hat{p}(y_{i_2}) \int_{-\infty}^{\infty} x_{i_2}\hat{p}(x_{i_2})\ dx_{i_2}\ dy_{i_2} \cdots \\
& \sum_{i_g=0}^{d} \int_{-\infty}^{\infty} y_{i_g}\hat{p}(y_{i_g}) \int_{-\infty}^{\infty} x_{i_g}\hat{p}(x_{i_g})\ dx_{i_g}\ dy_{i_g} \\
= & \left[\sum_{i=0}^{d} \int_{-\infty}^{\infty} y_i\hat{p}(y_i) \int_{-\infty}^{\infty} x_i\hat{p}(x_i)\ dx_i\ dy_i \right]^g \\
= & \left[\sum_{i=0}^{d} \kappa_i \right]^g
\end{aligned}
$$

where (here the second variable is introduced)

$$
\kappa_i = \begin{cases} I_{y_i} & \text{if } y_i \text{ is missings} \\ y_i \cdot \omega_i & \text{otherwise} \end{cases}
$$

and

$$
\omega_i = \begin{cases} I_{x_i} & \text{if } x_i \text{ is missings} \\ x_i & \text{otherwise} \end{cases}
$$

The integrals have the following solution

$$
I_{y_i} = \omega_i \int_{-\infty}^{\infty} y_i\hat{p}(y_i)\ dy_i = \omega_i \frac{1}{(n-m_i)} \sum_{\substack{k=1 \\ k \notin M_i}}^{n} y_{ki}
$$

$$
I_{x_i} = \int_{-\infty}^{\infty} x_i\hat{p}(x_i)\ dx_i = \frac{1}{(n-m_i)} \sum_{\substack{k=1 \\ k \notin M_i}}^{n} x_{ki}
$$

where m_i is the number of missings in the i-th variable and $M_i = \{i' : i' \in \{1..n\},\ x_{i'i} \text{ is a missing}\}$.

The integral of the form

$$
\int_{-\infty}^{\infty} x \cdot \exp\big(-a(y-x)^2 \big)\ dx
$$

is solved taking $t = y - x$, $dt = \frac{\partial}{\partial x}(y-x) = -dx$. Due to the substitution, the limits of integration are changed and it results in

$$-\int_{\infty}^{-\infty} (y-t) \cdot \exp(-at^2)\ dt$$

that is the same as

$$\int_{-\infty}^{\infty} (y-t) \cdot \exp(-at^2)\ dt = y \int_{-\infty}^{\infty} \exp(-at^2)\ dt - \int_{-\infty}^{\infty} t \cdot \exp(-at^2)\ dt = y\frac{\sqrt{\pi}}{\sqrt{a}} + 0$$

when $a > 0$ (in this case is so by definition). For more details, please check 10.4.

10.4 Solutions for the used integrals

The integral of the form

$$I(a) = \int_{-\infty}^{\infty} \exp(-at^2)\ dt$$

can be solved in this way

$$\begin{aligned}(I(a))^2 &= \int_{-\infty}^{\infty} \exp(-at^2)\ dt \int_{-\infty}^{\infty} \exp(-at'^2)\ dt' \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\Big(-a\big(t^2+t'^2\big)\Big)\ dtdt' \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\Big(-ar^2\Big)\ dtdt'\end{aligned}$$

where $r^2 = t^2 + t'^2$ is the circle equation with radius r. To be noticed, that the integrand presents the same value for all the points (t, t') that form the circle with radius r. Therefore, if the integrated plane is divided into several circles with $r = 0..\infty$, instead of thinking on it as a set of rectangles, is obtained then that the contribution of each annular region is $\exp(-ar^2) \cdot 2\pi r \cdot dr$, where dr is the width of every such strip. Thus

$$(I(a))^2 = \int_{0}^{\infty} \exp(-ar^2)\ 2\pi r\ dr$$

Be $u = r^2$, hence, $du = 2r \cdot dr$ and

$$(I(a))^2 = \pi \int_0^\infty \exp(-au)\ du = \frac{\pi}{a}$$

with $\int_0^\infty \exp(-au)\ du = -\frac{1}{a}\Big[\exp(-au)\Big]_0^\infty = \frac{1}{a}$. Therefore

$$I(a) = \sqrt{\frac{\pi}{a}}$$

The integral of the form

$$\int \exp(-at^2)\ dt$$

can be solved using u-substitution taking $at^2 = u^2$, hence $t = \frac{u}{\sqrt{a}}$ (with $a > 0$). Thus $dt = \frac{\partial}{\partial u}(\frac{u}{\sqrt{a}}) = \frac{1}{\sqrt{a}}du$ and

$$\int \exp(-at^2)\ dt = \frac{1}{\sqrt{a}} \int \exp(-u^2)\ du$$

where using error function $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\ dt$

$$\int \exp(-at^2)\ dt = \frac{\sqrt{\pi}}{2\sqrt{a}} \operatorname{erf}(t\sqrt{a})$$

$$\int \exp(-a(x-y)^2)\ dx = \int \exp(-at^2)\ dt = \frac{\sqrt{\pi}}{2\sqrt{a}} \operatorname{erf}((x-y)\sqrt{a})$$

taking $t = x - y$, $dt = \frac{\partial}{\partial x}(x-y) = dx$.

$$\int x \exp(-a(x-y)^2)\ dx = -\frac{1}{2a}\exp(-a(x-y)^2) + \frac{y\sqrt{\pi}}{2\sqrt{a}} \operatorname{erf}((x-y)\sqrt{a})$$

taking $t = x - y$, $dt = \frac{\partial}{\partial x}(x-y) = dx$.

$$\begin{aligned}\int x^2 \exp(-a(x-y)^2)\ dx =& 2y \int t \exp(-at^2)\ dt + y^2 \int \exp(-at^2)\ dt + \int t^2 \exp(-at^2)\ dt \\ =& \frac{\sqrt{\pi}}{4\sqrt{a^3}} \operatorname{erf}((x-y)\sqrt{a}) - \frac{x}{2a}\exp(-a(x-y)^2) \\ &- \frac{y}{2a}\exp(-a(x-y)^2) + \frac{y^2\sqrt{\pi}}{2\sqrt{a}} \operatorname{erf}((x-y)\sqrt{a})\end{aligned}$$

taking $t = x - y$, $dt = \frac{\partial}{\partial x}(x-y) = dx$ and $x = t + y$.

$$\int \exp(-a(x-y)^2)\ dy = -\int \exp(-at^2)\ dt = -\frac{\sqrt{\pi}}{2\sqrt{a}}\mathrm{erf}\left((x-y)\sqrt{a}\right)$$

taking $t = x - y$, $dt = \frac{\partial}{\partial y}(x-y) = -dy$.

$$\begin{aligned}\int \mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy =& y\mathrm{erf}\left((x-y)\sqrt{a}\right) + \frac{2\sqrt{a}}{\sqrt{\pi}}\int y\exp(-a(x-y)^2)\ dy \\ =& y\mathrm{erf}\left((x-y)\sqrt{a}\right) - \frac{1}{\sqrt{\pi}\sqrt{a}}\exp(-a(x-y)^2) - x\mathrm{erf}\left((x-y)\sqrt{a}\right)\end{aligned}$$

taking $u = \mathrm{erf}\left((x-y)\sqrt{a}\right)$, $du = \frac{\partial}{\partial y}\mathrm{erf}\left((x-y)\sqrt{a}\right) = -\frac{2\sqrt{a}}{sqrt\pi}\exp(-a(x-y)^2)\ dy$, $dv = dy$ and $v = \int\ dy = y$.

$$\begin{aligned}\int y\exp(-a(x-y)^2)\ dy =& \int t\exp(-at^2)\ dt - x\int \exp(-at^2)\ dt \\ =& -\frac{1}{2a}\exp(-a(x-y)^2) - \frac{x\sqrt{\pi}}{2\sqrt{a}}\mathrm{erf}\left((x-y)\sqrt{a}\right)\end{aligned}$$

taking $t = x - y$ and $dt = \frac{\partial}{\partial y}(x-y) = -dy$.

$$\begin{aligned}\int y\mathrm{erf}\left((x-y)\sqrt{a}\right)\ dy =& \frac{y^2}{2}\mathrm{erf}\left((x-y)\sqrt{a}\right) + \frac{\sqrt{a}}{\sqrt{\pi}}\int y^2\exp(-a(x-y)^2)\ dy \\ =& \frac{y^2}{2}\mathrm{erf}\left((x-y)\sqrt{a}\right) - \frac{x}{\sqrt{\pi}\sqrt{a}}\exp(-a(x-y)^2) \\ &- \frac{x^2}{2}\mathrm{erf}\left((x-y)\sqrt{a}\right) - \frac{1}{4a}\mathrm{erf}\left((x-y)\sqrt{a}\right) \\ &+ \frac{1}{2\sqrt{\pi}\sqrt{a}}\exp(-a(x-y)^2)(x-y)\end{aligned}$$

taking $u = \mathrm{erf}\left((x-y)\sqrt{a}\right)$, $du = -\frac{2\sqrt{a}}{\sqrt{\pi}}\exp(-a(x-y)^2)\ dy$, $dv = y$ and $v = \int y\ dy = \frac{y^2}{2}$.

$$\begin{aligned}\int y^2\exp(-a(x-y)^2)\ dy =& 2x\int t\exp(-at^2)\ dt - x^2\int \exp(-at^2)\ dt - \int t^2\exp(-at^2)\ dt \\ =& -\frac{x}{a}\exp(-a(x-y)^2) - \frac{x^2\sqrt{\pi}}{2\sqrt{a}}\mathrm{erf}\left((x-y)\sqrt{a}\right) \\ &- \frac{\sqrt{\pi}}{4\sqrt{a^3}}\mathrm{erf}\left((x-y)\sqrt{a}\right) + \frac{1}{2a}\exp(-a(x-y)^2)(x-y)\end{aligned}$$

taking $t = x - y$, $dt = \frac{\partial}{\partial y}(x-y) = -dy$ and $y = x - t$.

$$\begin{aligned}\int y^2\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy =&\frac{1}{3}\Big(y^3\mathrm{erf}\left((x-y)\sqrt{a}\right)-\frac{y^2}{\sqrt{\pi}\sqrt{a}}\exp(-a(x-y)^2)\\&-xy^2\mathrm{erf}\left((x-y)\sqrt{a}\right)+\frac{2}{\sqrt{\pi}\sqrt{a}}\int y\exp(-a(x-y)^2)\,dy\\&+2x\int y\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy\Big)\end{aligned}$$

taking $u=y^2$, $du=\frac{\partial}{\partial y}(y^2)=2y\ dy$, $dv=\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy$ and $v=\int\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy$.

$$\int y^3\exp(-a(x-y)^2)\,dy=-\frac{\sqrt{\pi}}{2\sqrt{a}}y^3\mathrm{erf}\left((x-y)\sqrt{a}\right)+\frac{3\sqrt{\pi}}{2\sqrt{a}}\int y^2\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy$$

taking $u=y^3$, $du=\frac{\partial}{\partial y}(y^3)=3y\ dy$, $dv=\exp(-a(x-y)^2)\,dy$ and $v=\int\exp(-a(x-y)^2)\,dy=-\frac{\sqrt{\pi}}{2\sqrt{a}}\mathrm{erf}\left((x-y)\sqrt{a}\right)$.

$$\begin{aligned}\int y^3\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy =&\frac{1}{4}\Big(y^4\mathrm{erf}\left((x-y)\sqrt{a}\right)-\frac{y^3}{\sqrt{\pi}\sqrt{a}}\exp(-a(x-y)^2)\\&-xy^3\mathrm{erf}\left((x-y)\sqrt{a}\right)+\frac{3}{\sqrt{\pi}\sqrt{a}}\int y^2\exp(-a(x-y)^2)\,dy\\&+3x\int y^2\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy\Big)\end{aligned}$$

taking $u=y^3$, $du=\frac{\partial}{\partial y}(y^3)=3y^2\ dy$, $dv=\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy$ and $v=\int\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy$.

$$\begin{aligned}\int y^4\mathrm{erf}\ \,dy =&\frac{1}{5}\Big(y^5\mathrm{erf}\left((x-y)\sqrt{a}\right)-\frac{y^4}{\sqrt{\pi}\sqrt{a}}\exp(-a(x-y)^2)-xy^4\mathrm{erf}\left((x-y)\sqrt{a}\right)\\&+\frac{4}{\sqrt{\pi}\sqrt{a}}\int y^3\exp(-a(x-y)^2)\,dy+4x\int y^3\mathrm{erf}\left((x-y)\sqrt{a}\right)\,dy\Big)\end{aligned}$$

The integral $\int t^2\exp(-at^2)\,dt$ is convenient to solve in this way:

$$\int t^2\exp(-at^2)\,dt=-\frac{\partial}{\partial a}\left(\int\exp(-at^2)\,dt\right)$$

Should be applied the derivative of the summation and product and, then, to apply the fundamental theorem of algebra (for the error function, that is not zero because the variable a appears in its upper limit)

$$
\begin{aligned}
-\frac{\partial}{\partial a}\left(\int \exp(-at^2)\ dt\right) &= -\frac{\partial}{\partial a}\left(\frac{\sqrt{\pi}}{2\sqrt{a}}\operatorname{erf}\left(t\sqrt{a}\right)\right) \\
&= -\left[\Big(\frac{\partial}{\partial a}\frac{\sqrt{\pi}}{2\sqrt{a}}\Big)\operatorname{erf}\left(t\sqrt{a}\right) + \frac{\sqrt{\pi}}{2\sqrt{a}}\Big(\frac{\partial}{\partial a}\operatorname{erf}\left(t\sqrt{a}\right)\Big)\right] \\
&= \frac{\sqrt{\pi}}{4\sqrt{a^3}}\operatorname{erf}\left(t\sqrt{a}\right) - \frac{1}{2a}\exp(-at^2)t
\end{aligned}
$$

11 References

[1] Consultations with my teacher, Lluís A. Belanche Muñoz.

[2] G. Nebot and Ll. Belanche. A kernel extension to handle missing data. In Bramer, Ellis and Petridis, editors, Research and Development in Intelligent Systems XXVI, Springer, 2010.

[3] Abramowitz M., Stegun I. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, New York: Dover 1965, ISBN $978 - 0486612720$.

[4] Ripley, B. Pattern Recognition and Neural Networks. Cambridge Univ. Press, (1996).

[5] V. Kobayashi, T. Aluja, L. Belanche. Handling missing values in kernel methods with application to microbiology data. 2014.

[6] Gibergans J., Gil A., Rovira C. Estadística. UOC, 4a ed. (2009).

[7] Bandwidth selection. Retrieved on 1st of October of 2014 from *https://stat.ethz.ch/R-manual/R-devel/library*

[8] **R** Documentation. Retrieved on 1st of December of 2014 from *http://cran.r-project.org/*

[9] σ estimation. Retrieved on 1st of October of 2014 from *http://cran.r-project.org/web/ packages/kernlab/*

[10] Notes on KDE - Manchester School of Mathematics. Retrieved from *http://www.maths.manchester.ac.uk/ peterf/MATH38011/*

[11] E. Parzen. On the estimation of a probability density function and mode. Annals of Mathematical Statistics, 33:1065–1076, 1962.

[12] [Bishop2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* 2006. Edicions Springer.

[13] J Shawe-Taylor, N Cristianini. *Kernel methods for pattern analysis.* 2004. Cambridge university press.

[14] A. Gelman, J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* 2006.

[15] S. van Buuren and K. Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. Journal of Statistical Software, 45(3):1–67, 2011.

[16] A. Rogier T. Donders, Geert J.M.G. van der Heijden, Theo Stijnen, Karel G.M. Moons. A gentle introduction to imputation of missing values. Journal of Clinical Epidemiology 59, 2006.