pharmaceutics

MDPI

Article

Predicting Pharmaceutical Particle Size Distributions Using Kernel Mean Embedding

Daan Van Hauwermeiren [1,2,*], Michiel Stock [3], Thomas De Beer [2] and Ingmar Nopens [1]

1 BIOMATH - Department of data analysis and mathematical modelling, Ghent University, Coupure Links 653, 9000 Gent, Belgium; ingmar.nopens@ugent.be
2 Laboratory of Pharmaceutical Process Analytical Technology - Department of pharmaceutical analysis, Ghent University, Ottergemsesteenweg 460, 9000 Gent, Belgium; Thomas.debeer@ugent.be
3 KERMIT - Department of data analysis and mathematical modelling, Ghent University, Coupure Links 653, 9000 Gent, Belgium; Michiel.stock@ugent.be
* Correspondence: daan.vanhauwermeiren@ugent.be; Tel.: +32-9-264-61-96

Received: 27 January 2020; Accepted: 9 March 2020; Published: 16 March 2020

Abstract: In the pharmaceutical industry, the transition to continuous manufacturing of solid dosage forms is adopted by more and more companies. For these continuous processes, high-quality process models are needed. In pharmaceutical wet granulation, a unit operation in the ConsiGma^{TM}-25 continuous powder-to-tablet system (GEA Pharma systems, Collette, Wommelgem, Belgium), the product under study presents itself as a collection of particles that differ in shape and size. The measurement of this collection results in a particle size distribution. However, the theoretical basis to describe the physical phenomena leading to changes in this particle size distribution is lacking. It is essential to understand how the particle size distribution changes as a function of the unit operation's process settings, as it has a profound effect on the behavior of the fluid bed dryer. Therefore, we suggest a data-driven modeling framework that links the machine settings of the wet granulation unit operation and the output distribution of granules. We do this without making any assumptions on the nature of the distributions under study. A simulation of the granule size distribution could act as a soft sensor when in-line measurements are challenging to perform. The method of this work is a two-step procedure: first, the measured distributions are transformed into a high-dimensional feature space, where the relation between the machine settings and the distributions can be learnt. Second, the inverse transformation is performed, allowing an interpretation of the results in the original measurement space. Further, a comparison is made with previous work, which employs a more mechanistic framework for describing the granules. A reliable prediction of the granule size is vital in the assurance of quality in the production line, and is needed in the assessment of upstream (feeding) and downstream (drying, milling, and tableting) issues. Now that a validated data-driven framework for predicting pharmaceutical particle size distributions is available, it can be applied in settings such as model-based experimental design and, due to its fast computation, there is potential in real-time model predictive control.

Keywords: granulation; wet granulation; continuous manufacturing; process modeling; particle size distributions; kernel methods; kernel mean embedding; predictive modeling; data-driven; machine learning

1. Introduction

In pharmaceutical twin-screw wet granulation (TSWG), a dry powder is granulated using a liquid to wet granules. The collection of granules differs in shape and size. Its measurement is called a particle size distribution (PSD). The measurement of this PSD is important to determine the input

setting of the TSWG. A fine balance is sought between aggregating the dry powder enough so that its flow properties are improved but not too much so that problems in the next unit operations arise. Too little aggregation results in dry powder that does not have good flow properties and could be blow out to the filters in the fluid bed dryer. Too much aggregation could lead to particles that are too large, needing longer drying times.

For the rapid development of a new formulation on the powder-to-tablet line under study (ConsiGma^{TM}-25), it is essential to have a predictive model linking the settings of the TSWG to the PSD. One possible approach is to use a population balance model (PBM), where the dynamical changes in particle size are described by making assumptions on how particles can aggregate and break in the TSWG [1].

This work takes a different look at the problem, using a data-driven approach to directly link the TSWG settings and the resulting PSD at the end of the granulator. The benefit this apprroach is that there is no need to make assumptions about the nature of the PSD itself. This is in contrast to the PBM framework, where thorough knowledge of the dynamics of aggregation and breakage of particles is essential. In Section 2, a general overview of the theory used in this work is given before diving into the mathematical descriptions of all the ideas in Section 3. In Section 4, the experimental set up and data collection are described. Section 5 describes the calibration procedure of the data-driven model, for which the results are presented and discussed in Section 6. General conclusions on this work are drawn in Section 7, and Section 8 lists some potential future research topics.

2. General Principles

In this section, a high-level overview explaining the general principles of the methodology in this paper is presented to give the reader an overview of the whole approach before diving into the details.

Supervised machine learning models learn a mapping from an arbitrary input to an arbitrary output space. How does one make a predictive model of a distribution of particle size as a function of the machine settings? One possible approach would be to aggregate the information contained in the whole distribution into a mean particle size, or an indication of the size of the largest or smallest particles. Typically, d_{10}, d_{50}, and d_{90} are used to describe the particle size in experimental papers in this application field, as in the work of Verstraeten et al. [2]. This approach is sensible if the underlying distribution is known. For instance, if the particle distribution is adequately approximated by a Gaussian distribution, then information on the mean particle size and the standard deviation is enough to fully characterize the distribution. However, in this application there is no knowledge of an underlying theoretical distribution to describe particle size. Hence, we need a framework that is agnostic with respect to the potential distribution types and which can link these with the process parameters.

To model distributions, we have to be able to construct manageable numerical representations. For example, we can compute all the moments of the distribution. In the jargon of the machine learning field, this would be called "feature generation" or a "feature map". Knowing all the moments of the distribution, the distribution itself is completely characterized. It is not convenient to work with an infinite number of moments if we want to link that to process settings.

Fortunately, there is a way to translate a distribution into a point in an implicit feature space in such a way that all information is retained. This procedure is explained in Section 3.2. This translation into a new space does not require the explicit calculation of a large number of features. It expresses all operations in terms of an inner product between pairs of data points in a feature space. These inner products are calculated using a class of functions called kernel functions. This approach of bypassing the generation of a large number of features and only using inner products is the core of a class of algorithms called kernel methods. A brief introduction to learning with kernels is written in Section 3.1. This transformation should be interpreted as a mapping to a mean in that feature space, hence the name of this technique is kernel mean embedding (KME).

Next, this theory is extended to conditional distributions (i.e., a PSD given some TSWG input settings in Section 3.3). If we have a way to deal with conditional distributions, the next logical step is to derive a learning framework—more specifically, structured output prediction. This framework makes it possible to learn the relationship between the mean embedding of a distribution in a feature space and the TSWG settings. To test if this approach can be generalized in the design space, a leave-one-out cross-validation is performed. The details to derive the learning theory are described in Section 3.4.

This work presents a way to manipulate distributions in such a way that all information is maintained and can be put this into a framework where the relation between the process parameters and the distribution can be learnt. The upside is that this framework allows for easy implementation of cross-validation so that the quality of the learnt relation can be assessed. However, the distributions are still expressed in that new feature space, which is not convenient to interpret. Section 3.5 deals with inverse operation from that feature space to a PSD, that is, recovering the function from the embedding.

In summary, the theory makes it possible to learn the (cross-validated) relationship between an input space and distributions and still maintain interpretability by allowing the inverse transformation to a distribution at the end. For a visual cue, this whole procedure is summarized in Figure 1.

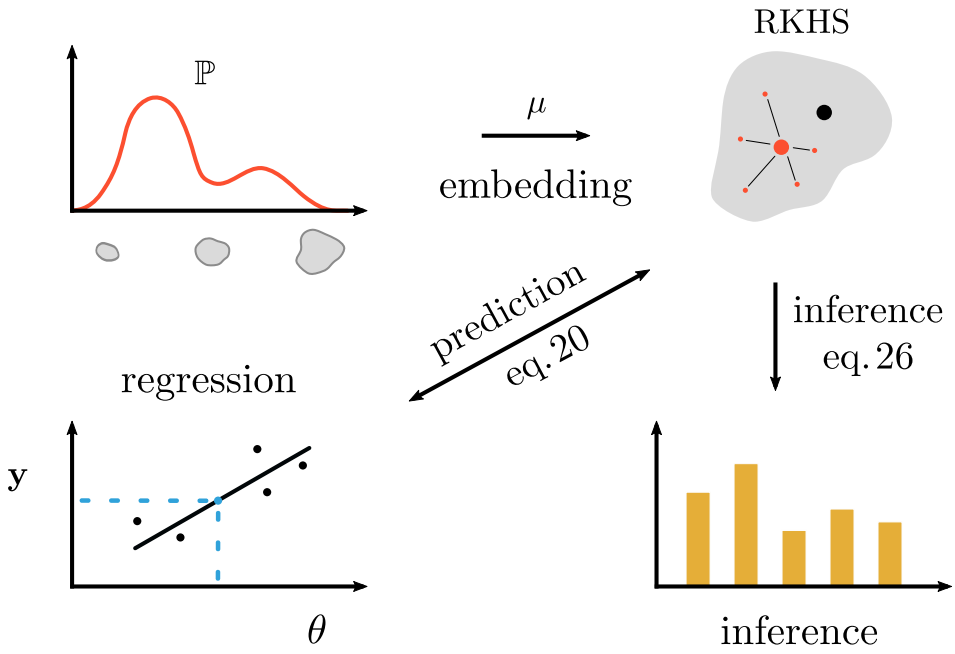


Figure 1. Visual representation of the general modeling principle in this paper. We start at the top left with our data: a measured particle size distribution. This distribution is translated into a new feature space through a kernel function φ. On the top right of the figure, the measured distribution is represented as a mean of features (large red dot), which might be slightly different from the embedding of the true distribution (large black dot). From this embedding we can perform inference or regression with the machine input settings. RKHS: recurrent kernel Hilbert space.

3. Theoretical Background

In the field of machine learning, the concept of the KME of distributions is a class of non-parametric methods in which a probability distribution is mapped to an element of a recurrent kernel Hilbert space (RKHS). These methods are a generalization of the classical feature mapping in kernel methods, which use individual data points. The learning framework is general in the sense that it can be applied to arbitrary distributions over any space on which a sensible kernel function can be defined. The focus of this work is on representing the distribution as real-valued scalars (i.e., particle sizes). However, various kernels have been proposed for other data types, such as strings, graphs, manifolds, and dynamical systems [3,4].

This part consists of five subsections: first, a general introduction to kernel methods is given. Next, the Hilbert space embedding of marginal and conditional distributions are discussed, which leads to the formulation of a framework for learning on distributional data. Finally, a method to recover distributions from Gaussian RKHS embeddings is discussed. In this work, we only discuss the relevant background information to understand the applications in this work. For an in-depth review of KME, its properties, and its applications, the reader is referred to the review article by Muandet et al. [5]. More details on the backbone of KME: RKHS, learning with kernels, and probability theory can be found in Hofmann et al. [4], Schölkopf et al. [6], Berlinet and Thomas-Agnan [7], respectively.

3.1. Learning with Kernels

The classical machine learning algorithms perceptron [8], support vector machine [9], and principal component analysis [10,11] consider the data, $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, with $\mathcal{X}$ a non empty set, through their inner product $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$. This inner product can be interpreted as a similarity measure between the elements of $\mathcal{X}$. This class of linear functions may be too restrictive for many applications if more complex relations between input and output data are sought. The core of kernel methods is to replace the inner product $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$ with another (non-linear) similarity measure. As an example, one can explicitly apply a non-linear transformation:

$$\begin{aligned} \phi : \mathcal{X} &\to \mathcal{F} \\ \boldsymbol{x} &\mapsto \phi(\boldsymbol{x}) \end{aligned} \tag{1}$$

from $\mathcal{X}$ to the high-dimensional feature space $\mathcal{F}$ and evaluate the inner product in the newly constructed space:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{F}}, \tag{2}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is the inner product of $\mathcal{F}$, ϕ is the feature map, and k is the kernel function which defines a non-linear similarity measure between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Given a learning algorithm that operates on the data through the inner product $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$, a non-linear extension of the algorithm can be made by substituting $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$ with $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{F}}$. The principles of the algorithms do not change, only the space in which the algorithms operate. The complexity of the algorithm is controlled by the complexity of the non-linear transformation ϕ. The evaluation of Equation (2) requires two steps: explicitly constructing the feature maps $\phi(\boldsymbol{x})$ and subsequently evaluating the inner product $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{F}}$. Issues can arise when $\phi(\boldsymbol{x})$ defines a transformation to a high-dimensional feature space. However, it is possible to evaluate $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{F}}$ directly without explicitly constructing the feature maps. This is an essential part of kernel methods, and in the machine learning community this is called "the kernel trick". A visual aid for this kernel trick is shown in Figure 2.

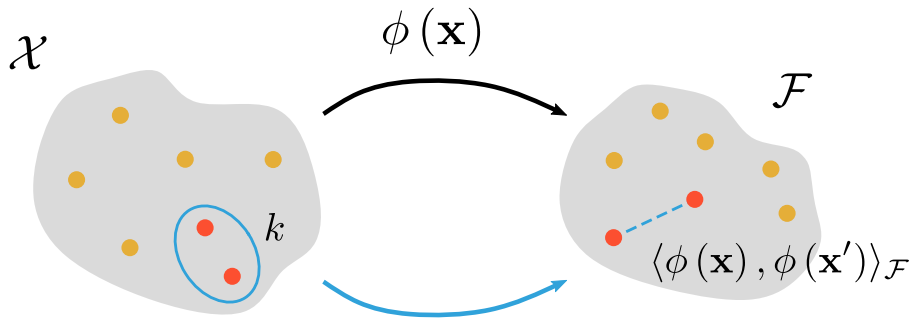


Figure 2. Visual representation of the kernel trick. The value of the kernel function of a pair of objects (denoted in red) in object space $\mathcal{X}$ is identical to an inner product of the representations of the objects in the implied Hilbert space $\mathcal{F}$.

This kernel trick can only be applied if k is positive definite. The positive definite kernel function $k(\boldsymbol{x}, \boldsymbol{x'})$ is central to the successful application of KME. This kernel function initially arises as a way to perform an inner product $\langle \boldsymbol{x}, \boldsymbol{x'} \rangle$ in a high-dimensional feature space $\mathscr{H}$ for some data points $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{X}$. The collection of all pairwise inner products within the set of data vectors $\boldsymbol{x}$ is called the $n \times n$-Gram or kernel matrix $K_{ij} := k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. In general, a symmetric function k, is called a positive definite kernel on $\mathcal{X}$ if the Gram matrix is positive definite, that is,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0, \quad \forall \boldsymbol{x}_i \in \mathcal{X}. \tag{3}$$

Equation (3) holds for any $n \in \mathbb{N}$, all finite sequences of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ in $\mathcal{X}$ and all choices of n real-valued coefficients $c_1, \ldots, c_n \in \mathbb{R}$ [12]. The positive definiteness of the kernel guarantees the existence of a dot product space $\mathcal{F}$ and a feature map $\phi : \mathcal{X} \to \mathcal{F}$ such that $k(\boldsymbol{x}, \boldsymbol{x'}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x'}) \rangle_{\mathcal{F}}$ [13] without needing to compute ϕ explicitly [6,9,14,15]. Moreover, a positive definite kernel induces a space of functions from $\mathcal{X}$ to $\mathbb{R}$ called an RKHS $\mathscr{H}$, hence also called a reproducing kernel [13]. An RKHS has two important properties: first, for any $\boldsymbol{x} \in \mathcal{X}$, the function $k(\boldsymbol{x}, \cdot) : \boldsymbol{y} \mapsto k(\boldsymbol{x}, \boldsymbol{y})$ is an element of $\mathscr{H}$. That is, whenever the kernel k is used, the feature space $\mathcal{F}$ is essentially the RKHS $\mathscr{H}$ associated with this kernel and it can be interpreted as a canonical feature map:

$$\begin{aligned} k : \mathcal{X} &\to \mathscr{H} \subset \mathbb{R}^{\mathcal{X}} \\ \boldsymbol{x} &\mapsto k(\boldsymbol{x}, \cdot) \end{aligned} \tag{4}$$

where $\mathbb{R}^{\mathcal{X}}$ denotes the vector space of functions from $\mathcal{X}$ to $\mathbb{R}$. The second property is that an inner product in $\mathscr{H}$ satisfies the reproducing property, i.e., for all functions $f \in \mathscr{H}$ and $\boldsymbol{x} \in \mathcal{X}$,

$$f(\boldsymbol{x}) = \langle f, k(\boldsymbol{x}, \cdot) \rangle_{\mathscr{H}}. \tag{5}$$

In particular: $k(\boldsymbol{x}, \boldsymbol{x'}) = \langle k(\boldsymbol{x}, \cdot), k(\boldsymbol{x'}, \cdot) \rangle_{\mathscr{H}}$. Further details on how $\phi(\boldsymbol{x}) = k(\boldsymbol{x}, \cdot)$ can be derived directly from the kernel k can be found in Schölkopf et al. [6]. The kernel used in this work is a Gaussian kernel, which is part of a class of kernels with interesting properties called radial basis functions (RBFs):

$$k^{\mathrm{RBF}}(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x'}\|^2}{2\sigma^2} \right) \tag{6}$$

with $\sigma > 0$ a bandwidth parameter. For $\sigma \to \infty$, the Gram matrix of this kernel becomes a matrix of ones, for $\sigma \to 0$, it becomes an identity matrix. The former situation implies that all instances are the same, the latter implies that they are all completely unique. The RBF kernel is a stationary kernel: it can be described as a function of the difference of its inputs. The RBF kernel is also called a universal kernel because any smooth function can be represented with a high degree of accuracy, assuming we can find a suitable value of the bandwidth. More details on different classes of kernel functions and their application domains can be found in Genton [16]. For further details on the properties of the RKHS and important theorems such as Mercer's and Bochner's theorem, the reader is referred to Muandet et al. [5], Mercer [12], and Bochner [17], respectively.

3.2. Hilbert Space Embedding of Marginal Distributions

In KME the concept of a feature map ϕ is extended to the space of probability distributions through the mapping μ which defines the representer in $\mathscr{H}$ of any distribution $\mathbb{P}$:

$$\begin{aligned} \mu : M_+^1(\mathcal{X}) &\to \mathscr{H} \\ \mathbb{P} &\mapsto \int_{\mathcal{X}} k(\boldsymbol{x}, \cdot)\, d\mathbb{P}(\boldsymbol{x}), \end{aligned} \tag{7}$$

with $M_+^1(\mathcal{X})$ the space of probability measures over a measurable space $\mathcal{X}$ [7,18]. The above mapping is the kernel mean embedding that is considered in this work:

$$\phi(\mathbb{P}) = \mu_{\mathbb{P}} := \mathbb{E}_{X\sim\mathbb{P}}[k(X, \cdot)] = \int_{\mathcal{X}} k(\boldsymbol{x}, \cdot)\, d\mathbb{P}(\boldsymbol{x}). \tag{8}$$

A visual representation of this mean embedding is given in Figure 3. In essence, the distribution $\mathbb{P}$ is transformed into an element in the feature space $\mathscr{H}$, which is just an RKHS corresponding to the kernel positive definite kernel k. This element (i.e., the mean embedding $\mu_{\mathbb{P}}$) is the expected value in that feature space. Since $\mathbb{P}$ is a probability density distribution, the expected value can be written as an integral. The derivation, proof, and properties of Equation (8) can be found in Muandet et al. [5], Berlinet and Thomas-Agnan [7], and Smola et al. [18].

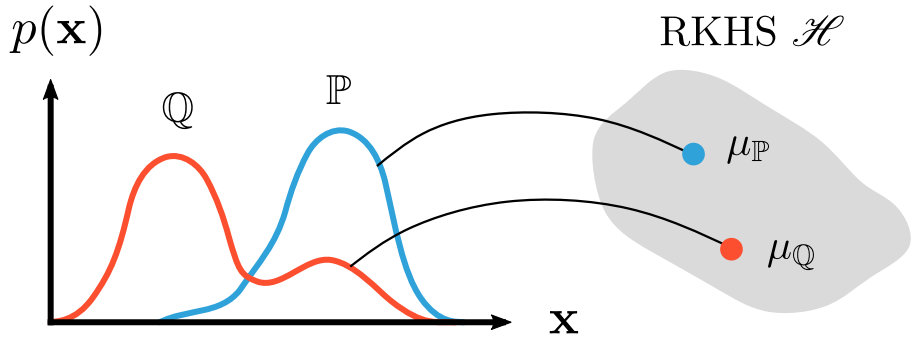


Figure 3. Embedding of marginal distributions $\mathbb{P}$ and $\mathbb{Q}$ into the RKHS $\mathscr{H}$ yielding $\mu_{\mathbb{P}}$ and $\mu_{\mathbb{Q}}$. Figure based on Muandet et al. [5].

Through Equation (8), most RKHS methods can therefore be extended to probability measures. When embedding a distribution in another space, it is crucial to understand what information of the distribution is retained by the kernel mean embedding. Consider the class of inhomogeneous polynomial kernels of order $p \in \mathbb{N}$:

$$k^{\text{poly}}(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + 1)^p = 1 + \binom{p}{1}\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + \binom{p}{2}\langle \boldsymbol{x}, \boldsymbol{x}' \rangle^2 + \cdots + \binom{p}{p}\langle \boldsymbol{x}, \boldsymbol{x}' \rangle^p. \tag{9}$$

Polynomial kernels of order p allow for learning a p-th order polynomial model w.r.t. the features. For our purposes, a polynomial kernel would model the p first moments of a distribution when used in KME. For a linear kernel, which is equal to computing the inner product, $\mu_{\mathbb{P}}$ equals the first moment of $\mathbb{P}$, whereas the polynomial kernel of order 2 allows the mean map to retain information on both the first and the second moments of $\mathbb{P}$. Generally speaking, the mean map using the inhomogeneous polynomial kernel of order p captures information up to the p-th moment of $\mathbb{P}$. Other explicit examples for some kernels can be found in Smola et al. [18], Fukumizu et al. [19], Sriperumbudur et al. [20], Gretton et al. [21], Schölkopf et al. [22]. There exists a class of kernel functions known as characteristic kernels for which the kernel mean representation captures all information about the distribution $\mathbb{P}$, with the Gaussian kernel used in this work as an example [23,24]. It follows that the RKHS endowed with the kernel k should contain a sufficiently rich class of functions to represent all higher-order moments of $\mathbb{P}$ [24]. The map $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective, implying that $\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathscr{H}} = 0$ if and only if $\mathbb{P} = \mathbb{Q}$, that is, $\mathbb{P}$ and $\mathbb{Q}$ are the same distribution. Injectivity of the map $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ makes the RKHS embedding suitable for regression problems since this map is inherently structurally identifiable (i.e., each element

in the feature space corresponds to one unique distribution in the original space). Lastly, it is necessary to point out that, in practice, access to the true distribution $\mathbb{P}$ is often lacking, and thereby the mean embedding $\mu_{\mathbb{P}}$ cannot be computed. Instead, often only an independent and identically distributed (iid) sample $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of the distribution is available. The standard estimator $\hat{\mu}_{\mathbb{P}}$ of the kernel mean $\mu_{\mathbb{P}}$ is an empirical average:

$$\hat{\mu}_{\mathbb{P}} := \frac{1}{n} \sum_{i=1}^{n} k\left(\boldsymbol{x}_i, \cdot\right), \tag{10}$$

with $\hat{\mu}_{\mathbb{P}}$ an unbiased estimate of $\mu_{\mathbb{P}}$. By the weak law of larger numbers, $\hat{\mu}_{\mathbb{P}}$ converges to $\mu_{\mathbb{P}}$ as $n \to \infty$ [25]. In this work, the data should be interpreted as a probability mass distribution associated with the sample $\mathbf{X}$. For example, $\hat{\mathbb{P}} := \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{x}_i}$, with $\delta_{\mathbf{x}}$ the Dirac measure defined for $\mathbf{x}$ in $\mathcal{X}$, such that the mean embedding takes the form of a weighted sum of feature vectors:

$$\hat{\mu}_{\mathbb{P}} := \sum_{i=1}^{n} w_i k\left(\boldsymbol{x}_i, \cdot\right), \tag{11}$$

with $\boldsymbol{w} = [w_i] \in \Delta^{n-1}$, that is, a histogram with weights $w_i > 0$ and subject to the constraint $\sum_i^n w_i = 1$ [26]. A comparison of Equations (8), (10), and (11) is visualized in Figure 4.

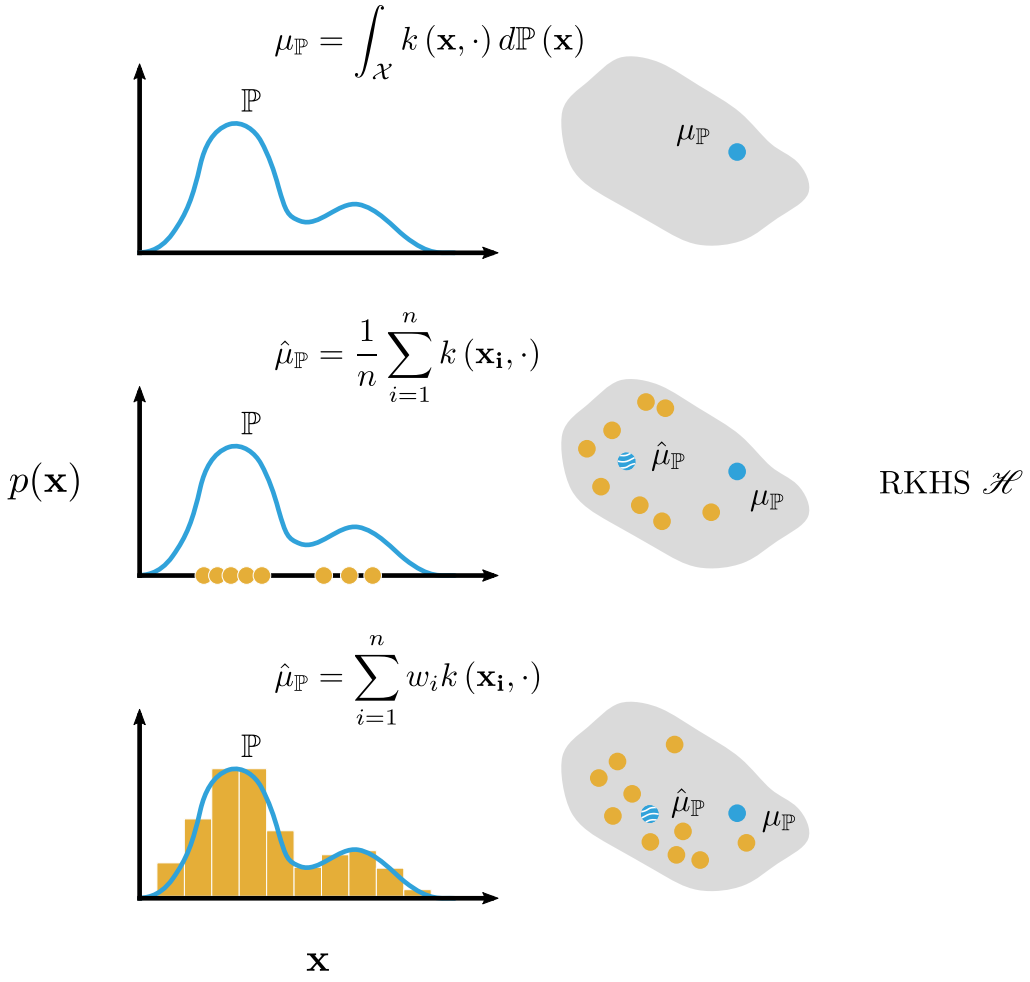


Figure 4. A comparison of the embedding of marginal distributions $\mathbb{P}$ into the RKHS $\mathscr{H}$: (**top**) a continuous distribution is an integral over the Hilbert space, (**middle**) a sample distribution is the arithmetic mean over the embeddings of the individual samples, and (**bottom**) a probability mass function is the weighted average over the individual embeddings.

In summary, the described framework allows the transformation of marginal distributions into a rich feature space without making assumptions about the underlying distribution (e.g., belonging to a particular class of distributions) by using an extension of previously established kernel methods. By carefully choosing the nature of this transformation, all information on this distribution is retained. Next, it can be proven that this map is identifiable, which makes this representation suitable for regression problems. Finally, compared to density estimation approaches, the kernel mean representation is less prone to the curse of dimensionality [27–29].

3.3. Hilbert Space Embedding of Conditional Distributions

In the previous subsection, the fundamentals of the mean map for marginal distributions were laid out. In this subsection, the extension of kernel mean embedding to a conditional distribution $\mathbb{P}(Y|X)$ and $\mathbb{P}(Y|X = \boldsymbol{x})$ for some $\boldsymbol{x} \in \mathcal{X}$ is discussed [26,30]. The conditional distribution captures the functional relationship between the two random variables X and Y. Conditional mean embedding thus extends the capability of kernel mean embedding to model more complex dependence.

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be positive definite kernels for the domains of X and Y, respectively. The RKHSs associated with these kernels are $\mathscr{H}$ and $\mathscr{G}$. The conditional mean embeddings of the conditional distributions $\mathbb{P}(Y|X)$ and $\mathbb{P}(Y|X = \boldsymbol{x})$ can be written as $\mathcal{U}_{Y|X} : \mathscr{H} \to \mathscr{G}$ and $\mathcal{U}_{Y|\boldsymbol{x}} \in \mathscr{G}$, such that they satisfy:

$$\mathcal{U}_{Y|\boldsymbol{x}} = \mathbb{E}_{Y|\boldsymbol{x}}[\varphi(Y)|X = \boldsymbol{x}] = \mathcal{U}_{Y|X} k(\boldsymbol{x}, \cdot) \tag{12}$$

$$\mathbb{E}_{Y|\boldsymbol{x}}[g(Y)|X = \boldsymbol{x}] = \langle g, \mathcal{U}_{Y|\boldsymbol{x}} \rangle_{\mathscr{G}}, \quad \forall g \in \mathscr{G}. \tag{13}$$

$\mathcal{U}_{Y|X}$ is an operator from one RKHS $\mathscr{H}$ to the other RKHS $\mathscr{G}$, and $\mathcal{U}_{Y|\boldsymbol{x}}$ is an element in $\mathscr{G}$. Equation (12) states that the conditional mean embedding of the conditional distribution $\mathbb{P}(Y|X = \boldsymbol{x})$ corresponds to the conditional expectation of the feature map of Y given that $X = \boldsymbol{x}$. The operator $\mathcal{U}_{Y|X}$ is the conditioning operation that when applied to $\phi(\boldsymbol{x}) \in \mathscr{H}$ yields the conditional mean embedding $\mathcal{U}_{Y|\boldsymbol{x}}$. Equation (13) describes the reproducing property of $\mathcal{U}_{Y|\boldsymbol{x}}$, that is, it should be a representer of conditional expectation in $\mathscr{G}$ w.r.t. $\mathbb{P}(Y|X = \boldsymbol{x})$. Using the definition of Song et al. [26,30]: let $\mathcal{C}_{XX} : \mathscr{H} \to \mathscr{H}$ and $\mathcal{C}_{XY} : \mathscr{H} \to \mathscr{G}$ be the covariance operator on X and cross-covariance operator from X to Y, respectively. Then, the conditional mean embedding $\mathcal{U}_{Y|X}$ and $\mathcal{U}_{Y|\boldsymbol{x}}$ are defined as:

$$\mathcal{U}_{Y|X} := \mathcal{C}_{XY} \mathcal{C}_{XX}^{-1} \tag{14}$$

$$\mathcal{U}_{Y|\boldsymbol{x}} := \mathcal{C}_{XY} \mathcal{C}_{XX}^{-1} k(\boldsymbol{x}, \cdot). \tag{15}$$

A visual explanation of these concepts can be found in Figure 5.

Further, Fukumizu et al. [23,31] state that if $\mathbb{E}_{Y|X}[g(Y)|X = \cdot] \in \mathscr{H}$ for any $g \in \mathscr{G}$, then

$$\mathcal{C}_{XX} \mathbb{E}_{YX}[g(Y)|X = \cdot] = \mathcal{C}_{XY} g. \tag{16}$$

For some $\boldsymbol{x} \in \mathcal{X}$, by virtue of the reproducing property, we have that

$$\mathbb{E}_{Y|\boldsymbol{x}}[g(Y)|X = \boldsymbol{x}] = \langle \mathbb{E}_{Y|X}[g(Y)|X], k(\boldsymbol{x}, \cdot) \rangle_{\mathscr{H}}. \tag{17}$$

Combining Equations (16) and (17) and taking the conjugate transpose of $\mathcal{C}_{XX}^{-1} \mathcal{C}_{XY}$ yields

$$\mathbb{E}_{Y|\boldsymbol{x}}[g(Y)|X = \boldsymbol{x}] = \langle g, \mathcal{C}_{YX} \mathcal{C}_{XX}^{-1} k(\boldsymbol{x}, \cdot) \rangle_{\mathscr{G}} = \langle g, \mathcal{U}_{Y|\boldsymbol{x}} \rangle_{\mathscr{G}}. \tag{18}$$

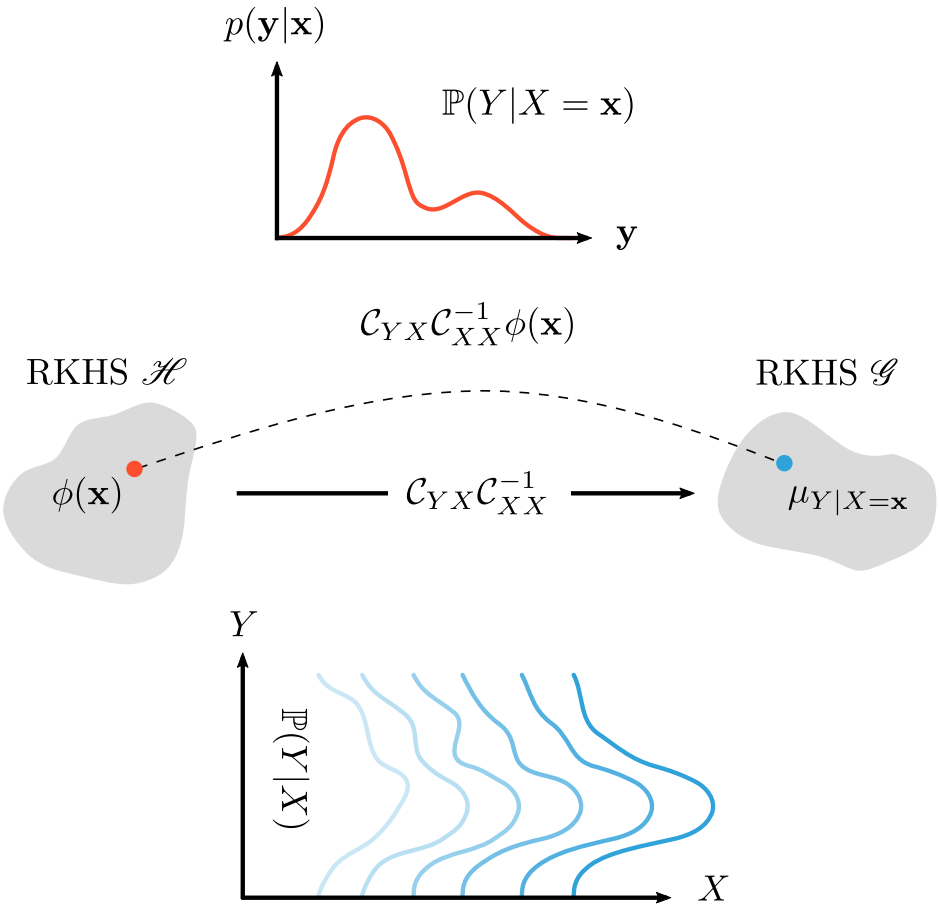


Figure 5. The embedding of conditional distribution $\mathbb{P}(Y|X)$ is not a single element in the RKHS. Instead, it may be viewed as a family of Hilbert space embeddings of the conditional distributions $\mathbb{P}(Y|X=\boldsymbol{x})$ indexed by the conditioning variable X. In other words, the conditional mean embedding can be viewed as an operator mapping from the RKHS $\mathscr{H}$ for features to RKHS $\mathscr{G}$ for distributions. Figure based on Muandet et al. [5].

It is important to note that the operator $\mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}$ may not exist in the continuous domain because the assumption that $\mathbb{E}_{YX}\left[g\left(Y\right)|X=\cdot\right] \in \mathscr{H}$, $\forall g \in \mathscr{G}$ may not hold in general [23,30]. To ensure existence, a regularised version of Equation (13) can be used, that is, $\mathcal{C}_{YX}\left(\mathcal{C}_{XX}+\lambda\mathcal{I}\right)^{-1}k\left(\boldsymbol{x},\cdot\right)$, where $\lambda > 0$ is a regularization parameter and $\mathcal{I}$ is the identity operator in $\mathscr{H}$. Fukumizu et al. [31] showed that under mild conditions, its empirical estimator is a consistent estimator of $\mathbb{E}_{Y|X}\left[g\left(Y\right)|X=\boldsymbol{x}\right]$.

In practice, some technical issues arise: since the joint distribution $\mathbb{P}\left(X,Y\right)$ is unknown, $\mathcal{C}_{XX}$ and $\mathcal{C}_{YX}$ cannot be computed directly. The solution is to rely on the iid sample $\left(\boldsymbol{x}_1,\boldsymbol{y}_1\right),\ldots,\left(\boldsymbol{x}_n,\boldsymbol{y}_n\right)$ from $\mathbb{P}\left(X,Y\right)$. Let $\Upsilon := \left[\phi(\boldsymbol{x}_1),\ldots,\phi(\boldsymbol{x}_n)\right]^T$ and $\Phi := \left[\varphi(\boldsymbol{y}_1),\ldots,\varphi(\boldsymbol{y}_n)\right]^T$ where $\phi : \mathcal{X} \to \mathscr{H}$ and $\varphi : \mathcal{Y} \to \mathscr{G}$ are the feature maps associated with the kernels k and l, respectively. The corresponding Gram matrices are defined as $\boldsymbol{K} = \Upsilon^T\Upsilon$ and $\boldsymbol{L} = \Phi^T\Phi$. Using the former definitions, the empirical estimator of the conditional mean embedding is given by

$$\begin{aligned}
\hat{\mathcal{C}}_{YX}\left(\hat{\mathcal{C}}_{XX}+\lambda\mathcal{I}\right)^{-1}k(\boldsymbol{x},\cdot) &= \frac{1}{n}\Phi\Upsilon^T\left(\frac{1}{n}\Upsilon\Upsilon^T+\lambda\mathcal{I}\right)^{-1}k(\boldsymbol{x},\cdot)\\
&= \Phi\Upsilon^T\left(\Upsilon\Upsilon^T+n\lambda\mathcal{I}\right)^{-1}k(\boldsymbol{x},\cdot)\\
&= \Phi\left(\Upsilon^T\Upsilon+n\lambda\boldsymbol{I}_n\right)^{-1}\Upsilon^Tk(\boldsymbol{x},\cdot)\\
&= \Phi\left(\boldsymbol{K}+n\lambda\boldsymbol{I}_n\right)^{-1}\boldsymbol{k}_{\boldsymbol{x}}. \qquad (19)
\end{aligned}$$

As derived by Song et al. [30], the conditional mean embedding of $\mu_{Y|\boldsymbol{x}}$ can be estimated using

$$\hat{\mu}_{Y|\boldsymbol{x}} = \Phi \left(\boldsymbol{K} + n\lambda \boldsymbol{I}_n\right)^{-1} \boldsymbol{k}_{\boldsymbol{x}}. \tag{20}$$

Let $\hat{\boldsymbol{\beta}}_\lambda := \left(\boldsymbol{K} + n\lambda \boldsymbol{I}_n\right)^{-1} \boldsymbol{k}_{\boldsymbol{x}} \in \mathbb{R}^n$, then Equation (20) can be written as $\hat{\mu}_{Y|\boldsymbol{x}} = \Phi\hat{\boldsymbol{\beta}}_\lambda = \sum_{i=1}^{n} (\hat{\boldsymbol{\beta}}_\lambda)_i \varphi(\boldsymbol{y}_i)$. It should be noted that this last equation is in a form similar to Equation (11). So, in conclusion, x determines the weights for the embedding of $\mathbb{P}(Y|\boldsymbol{x})$.

3.4. Learning on Distributional Data

Zhang et al. [32] and Grünewälder et al. [33] observed that the conditional mean embedding has a natural interpretation as a solution to the vector-valued regression problem. Recall that the conditional mean embedding is defined via $\mathbb{E}[g(Y)|X = \boldsymbol{x}] = \langle g, \hat{\mu}_{Y|\boldsymbol{x}} \rangle_{\mathcal{G}}$. That is, for every $\boldsymbol{x} \in \mathcal{X}$, $\hat{\mu}_{Y|\boldsymbol{x}}$ is a function on $\mathcal{Y}$ and thereby defines a mapping from $\mathcal{X}$ to $\mathcal{G}$. Furthermore, the empirical estimator in Equation (20) can be expressed as $\hat{\mu}_{Y|\boldsymbol{x}} = \Phi \left(\boldsymbol{K} + n\lambda \boldsymbol{I}_n\right)^{-1} \boldsymbol{k}_{\boldsymbol{x}}$, which already suggests that the conditional mean embedding is the solution to an underlying regression problem. Given an iid sample $(\boldsymbol{x}_1, \boldsymbol{z}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{z}_n) \in \mathcal{X} \times \mathcal{G}$, a vector-valued regression problem can be formulated as:

$$\hat{\mathcal{E}}_\lambda(f) = \sum_{i=1}^{n} ||\boldsymbol{z}_i - f(\boldsymbol{x}_i)||_{\mathcal{G}}^2 + \lambda ||f||_{\mathcal{H}_\Gamma}^2, \tag{21}$$

where $\mathcal{G}$ is a Hilbert space, $\mathcal{H}_\Gamma$ denotes an RKHS of vector-valued functions from $\mathcal{X}$ to $\mathcal{G}$ and $\hat{\mathcal{E}}_\lambda$ is the error associated with this regression problem [34]. Grünewälder et al. [33] show that $\hat{\mu}_{Y|X}$ can be obtained as a minimizer of the optimization of Equation (21). A natural optimization problem for the conditional mean embedding is to find a function $\mu : \mathcal{X} \to \mathcal{G}$ that minimizes an objective. Grünewälder et al. [33] shows that that objective can be bounded from above by a surrogate loss function, which in its empirical counterpart is described as

$$\hat{\mathcal{E}}_s[\mu] = \sum_{i=1}^{n} ||l(\boldsymbol{y}_i, \cdot) - \mu(\boldsymbol{x}_i)||_{\mathcal{G}}^2 + \lambda ||\mu||_{\mathcal{H}_\Gamma}^2, \tag{22}$$

with an added regularization term to provide a well-posed problem and prevent overfitting. This vector-values regression interpretation of conditional mean embedding has the advantage that a cross-validation procedure for parameter or model selection can be used because the loss function is well-defined. Since the analysis is done under the assumption that $\mathcal{G}$ is finite-dimensional, the conditional mean embedding is simply the ridge regression of feature vectors. Given $\hat{\boldsymbol{\beta}}_\lambda := \left(\boldsymbol{K} + n\lambda \boldsymbol{I}_n\right)^{-1} \boldsymbol{k}_{\boldsymbol{x}}$, the hat matrix, $\boldsymbol{H}_\lambda$, in the ridge regression context is defined as:

$$\begin{aligned} \boldsymbol{H}_\lambda \boldsymbol{k}_{\boldsymbol{x}} &= \hat{\boldsymbol{k}}_{\boldsymbol{x}} \\ &= \Phi\hat{\boldsymbol{\beta}}_\lambda \end{aligned} \tag{23}$$

$$\boldsymbol{H}_\lambda = \boldsymbol{K} \left(\boldsymbol{K} + \lambda \boldsymbol{I}\right)^{-1}. \tag{24}$$

The estimated conditional embedding using leave-one-out cross-validation (LOOCV) is then defined as [35]:

$$\hat{\boldsymbol{\mu}}_{Y|\boldsymbol{x}}^{\text{LOOCV}} = \left(\boldsymbol{I} - \text{diag}\left(\boldsymbol{H}_\lambda\right)\right)^{-1} \left(\boldsymbol{H}_\lambda - \text{diag}\left(\boldsymbol{H}_\lambda\right)\right) \Phi, \tag{25}$$

with $\text{diag}(\cdot)$ denoting the diagonal matrix. Note that using Equation (25), it possible to calculate all LOOCV conditional embeddings at once using matrix multiplications. For the interpretability of the results, the underlying distribution needs to be recovered from the LOOCV conditional mean embedding. This is described in the next paragraph.

3.5. Recovering Distributions from RKHS Embeddings

Given a kernel mean embedding $\mu_{\mathbb{P}}$, is it possible to recover essential properties of $\mathbb{P}$ from $\mu_{\mathbb{P}}$? This problem is known in the literature as the distributional pre-image problem [36–38]. It is important to note that there is a distinction with the classical pre-image problem, which does not involve probability distributions [5]. In this problem, objects in the input space are sought which correspond with a specific KME in the feature space. In this way, meaningful information of an underlying distribution can be recovered from an estimate of its embedding. Let $\mathbb{P}_{\boldsymbol{\theta}}$ be an arbitrary distribution parametrized by $\boldsymbol{\theta}$ and let $\mu_{\mathbb{P}_{\boldsymbol{\theta}}}$ be its mean embedding in $\mathscr{H}$. $\mathbb{P}_{\boldsymbol{\theta}}$ can be found by the following minimization problem

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \|\hat{\mu}_Y - \mu_{\mathbb{P}_{\boldsymbol{\theta}}}\|^2_{\mathscr{H}} \\ &= \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \langle \hat{\mu}_Y, \hat{\mu}_Y\rangle - 2\langle \hat{\mu}_Y, \mu_{\mathbb{P}_{\boldsymbol{\theta}}}\rangle + \langle \mu_{\mathbb{P}_{\boldsymbol{\theta}}}, \mu_{\mathbb{P}_{\boldsymbol{\theta}}}\rangle,\end{aligned} \tag{26}$$

subject to appropriate constraints on the parameter vector $\boldsymbol{\theta}$. Equation (26) minimizes the maximum mean discrepancy (MMD), which is defined by the idea of representing distances between distributions as distances between mean embeddings of features. Applied to this work, $\hat{\mu}_Y$ should be interpreted as the estimated conditional embedding using LOOCV, defined by Equation (25). The term $\langle \hat{\mu}_Y, \hat{\mu}_Y\rangle$ is only a function of the estimated conditional embedding, thus is constant and is left out of the minimization. Assume that $\mu_{\mathbb{P}_{\boldsymbol{\theta}}} = \sum_{i=1}^{n} \alpha_i \varphi(\boldsymbol{y}_i)$ for some $\boldsymbol{\alpha} \in \Delta^{n-1}$, or in words: $\mathbb{P}_{\boldsymbol{\theta}}$ is a histogram. It follows that $\langle \mu_{\mathbb{P}_{\boldsymbol{\theta}}}, \mu_{\mathbb{P}_{\boldsymbol{\theta}}}\rangle = \boldsymbol{\alpha}'(\boldsymbol{L} + \lambda \boldsymbol{I})\boldsymbol{\alpha}$ with $\boldsymbol{L}_{ij} = l(\boldsymbol{y}_i, \boldsymbol{y}_j)$. The addition of a regularizing term λ allows us to cast the optimization as a standard quadratic programming problem. Finally, $\langle \hat{\mu}_Y, \mu_{\mathbb{P}_{\boldsymbol{\theta}}}\rangle$ is then equal to the dot product of $\mathbb{P}_{\boldsymbol{\theta}}$ and the conditional embedding of Equation (25). The optimization in Equation (26) can thus be written as

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}\in\Delta^{n-1}} \boldsymbol{\alpha}'(\boldsymbol{L} + \lambda \boldsymbol{I})\boldsymbol{\alpha} - 2\boldsymbol{\alpha} \cdot \hat{\boldsymbol{\mu}}^{\text{LOOCV}}_{Y|\boldsymbol{x}}. \tag{27}$$

Although it is possible to solve Equation (27) and find a distributional pre-image, it is not clear what kind of information of $\mathbb{P}$ this pre-image represents. Kanagawa and Fukumizu [38] considers the recovery of the information of a distribution from an estimate of the kernel mean when the Gaussian RBF kernel on Euclidean space is used. They show that under some situations certain statistics of $\mathbb{P}$ can be recovered, namely its moments and measures on intervals, from $\hat{\mu}_{\mathbb{P}}$, and that the density of $\mathbb{P}$ can be estimated from $\hat{\mu}_{\mathbb{P}}$ without any parametric assumption on $\mathbb{P}$ (Kanagawa and Fukumizu [38]; Theorem 2).

4. Experimental Set Up and Data Collection

The application field of this work is pharmaceutical manufacturing. More specifically, the data gathered for this study originate from the high-shear TSWG unit operation in the ConsiGma™-25 system (GEA Pharma systems, Collette, Wommelgem, Belgium) continuous powder-to-tablet line. A schematic representation of the production line is shown in Figure 6. A more in-depth depiction of the TSWG can be found in Figure 7, displaying the input and output data of the system. The experimental set up of this paper is described in the work of Verstraeten et al. [2]. Here, only a summary with the details relevant to this work is given. For the full details, the reader should refer to the aforementioned paper. The TSWG is comprised of two 25 mm diameter self-wiping, co-rotating screws with a length-to-diameter ratio of 20 : 1. The preblend and the granulation liquid (demineralized water) are introduced into the system by a gravimetric twin-screw loss-in-weight feeder (KT20, K-Tron Soder, Niederlenz, Switzerland), and two out-of-phase peristaltic pumps located on top of the granulator (Watson Marlow, Cornwall, UK), connected to 1.6 mm nozzles. In this work, the data for the hydrophobic model drug were used. The preblend for this model drug contains 60% (*w/w*) hydrochlorothiazide (UTAG, Almere, The Netherlands), 16% (*w/w*) lactose monohydrate (Lactochem® Regular, DFE Pharma, Goch, Germany), 16% (*w/w*) microcrystalline cellulose (Avicel® PH 101, FMC,

Philadelphia, PA, USA), 3% (*w/w*) hydroxypropylcellulose (Klucel® EXF, Ashland, Covington, KY, USA), and 5% (*w/w*) crosscarmellose sodium (Ac-Di-Sol®, FMC, Philadelphia, PA, USA). A three-level full-factorial experimental design was used to study the influence of the granulation process parameters screw speed (450, 675, and 900 rpm), material throughput (5, 12.5, and 25 kg/h), and liquid-to-solid ratio (0.3, 0.45, and 0.6). An overview of the process conditions of this experimental design is listed in Table 1. This experiment was performed with a fixed screw configuration: two kneading compartments, each comprised of six kneading elements (length = diameter/6 for each element) with a 60° stagger angle, separated by a conveying element with the same length equal to 1.5 times the diameter. The barrel's jacket temperature was set at 25 °C. The samples were collected at four locations inside the barrel, however, only the measurements at the end of the granulator are considered in this work. After collection, the samples were oven-dried before the measurement of the PSD and other properties. The size and shape distribution of the collected, oven-dried, granule samples were analyzed using a QICPIC particle size analyzer with WINDOX 5.4.1.0 software (Sympatec, GmbH, Clausthal-Zellerfeld, Germany). The number of bins in the data were taken from previous work on population balance models [1]. They were chosen such that experimental data of the PSD by QICPIC could be loaded swiftly without the need for interpolation. The grid was comprised of a total of 35 bins, logarithmically spaced between 8.46 µm and 6765.36 µm.

Table 1. Process parameters of the experimental conditions. L/S: liquid-to-solid ratio.

Experiment	Throughput (kg/h)	Screw Speed (rpm)	L/S Ratio (—)
N1	5	450	0.3
N2	5	675	0.3
N3	5	900	0.3
N4	12.5	450	0.3
N5	12.5	675	0.3
N6	12.5	900	0.3
N7	25	450	0.3
N8	25	675	0.3
N9	25	900	0.3
N10	5	450	0.45
N11	5	675	0.45
N12	5	900	0.45
N13	12.5	450	0.45
N14	12.5	675	0.45
N15	12.5	900	0.45
N16	25	900	0.45
N17	25	450	0.45
N18	5	900	0.6
N19	5	450	0.6
N20	12.5	450	0.6
N21	12.5	675	0.6
N22	12.5	900	0.6
N23	25	900	0.6
N24	25	675	0.6
N25	25	450	0.6
N26	12.5	675	0.375
N27	12.5	675	0.525
N28	12.5	675	0.337
N29	12.5	675	0.563

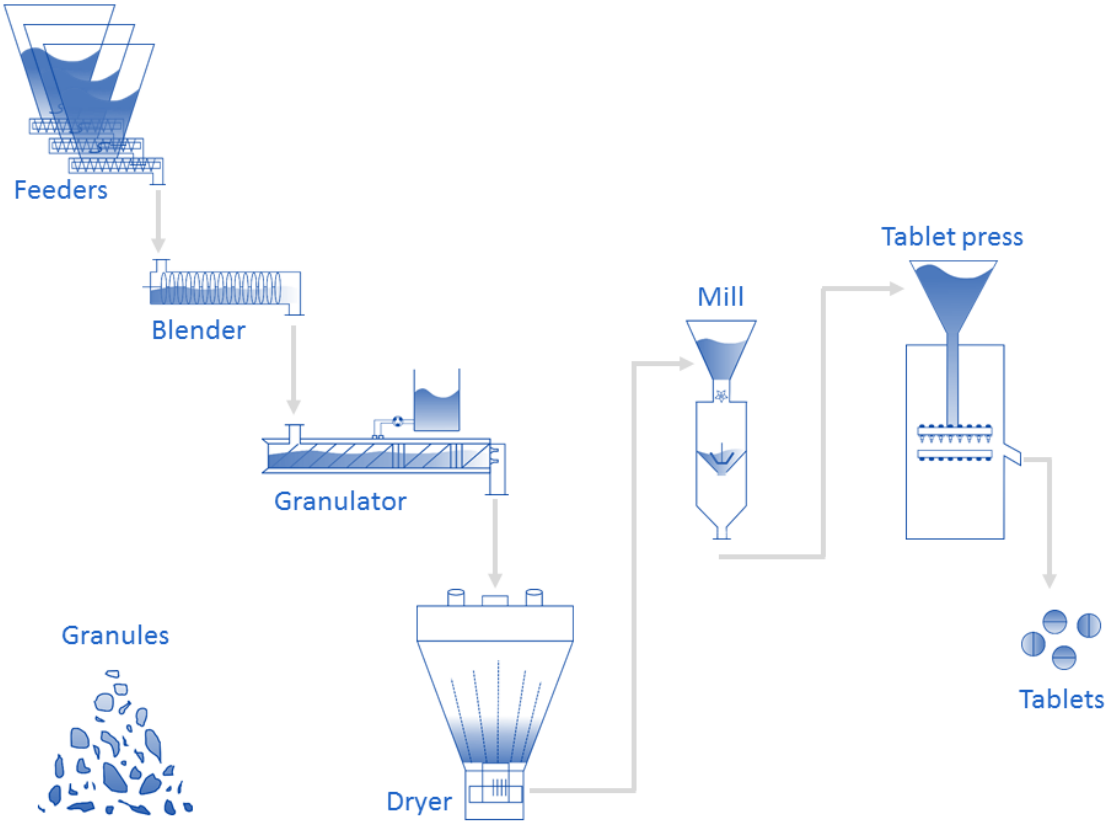


Figure 6. Schematic representation of the ConsiGma^{TM}-25 system (GEA Pharma systems, Collette, Wommelgem, Belgium) continuous powder-to-tablet line.

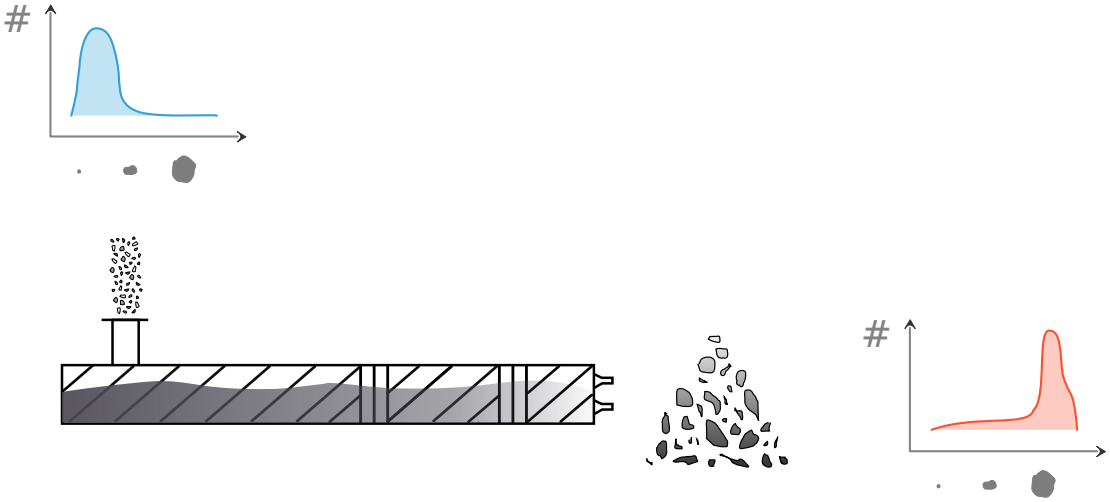


Figure 7. Schematic representation of the input and output data of the twin-screw wet granulation (TSWG), part of the ConsiGma^{TM}-25 system (GEA Pharma systems, Collette, Wommelgem, Belgium) continuous powder-to-tablet line.

5. Calibration Procedure

First, the process parameters $\boldsymbol{X}$ are standardized by removing the mean and scaling to unit variance. Next, some hyperparameters need to be defined: the bandwidth parameters σ of the RBF kernels and the regularization parameter λ. For the kernel on the grid of the distributions, σ is chosen

via the median heuristic [39]: $\sigma^2 = \text{median}\left\{\|\log_{10}(x_i) - \log_{10}(x_j)\|^2 : i, j = 1, \ldots, n\right\}$. Note that the logarithm of grid values is taken, as the grid spans more than three orders of magnitude. This brings the kernel values more closely together and gives more realistic results. For the kernel on the process parameters, the bandwidth is chosen as $\sigma^2 = 0.1$, which is approximately $^1/_{10}$ of the length scale. Finally, for numerical stability reasons (especially for the pre-imaging problem), a bias is added to the diagonal of the Gram matrix of both kernels: 0.05 for k and 0.1 for l. For a visual cue: the heatmaps of two Gram matrices of kernels are given in Figure 8. The regularization parameter λ is estimated via LOOCV: its value is altered so that the squared error between the mean embedding of the measured distributions and estimated distributions via LOOCV is minimized:

$$\lambda = \arg\min_{\lambda} \|\mu_{\mathbb{P}(Y|X)} - \hat{\mu}^{\text{LOOCV}}_{\mathbb{P}(Y|X)}\|^2_{\mathscr{G}}. \tag{28}$$

To assess the model quality, three different measures are used. In the following equations, $\mathbb{P}$ is the measured distribution, and $\hat{\mathbb{P}}$ is the estimated distribution using LOOCV KME. The MMD is calculated as shown before in Equation (26):

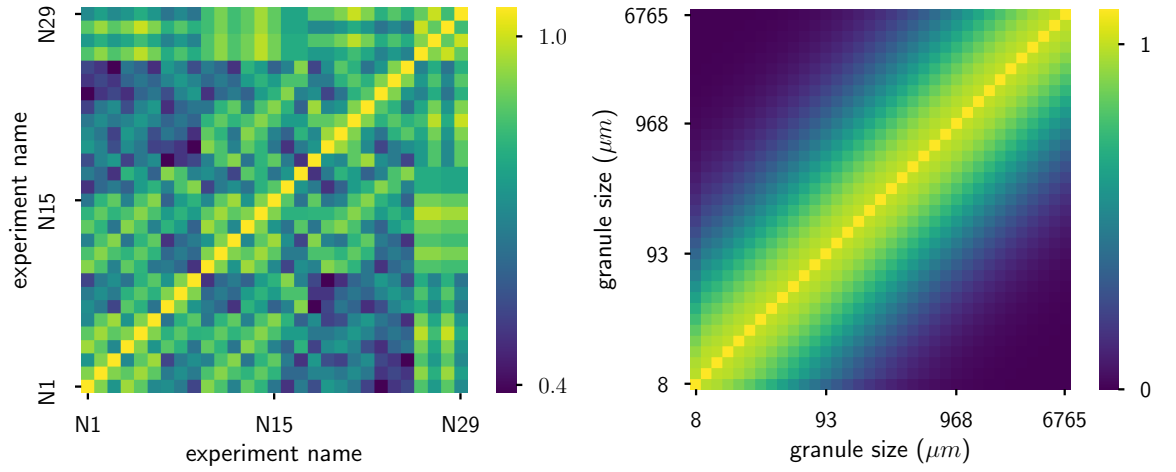
$$D_{\text{MMD}} = \|\mu_{\mathbb{P}} - \mu_{\hat{\mathbb{P}}}\|^2_{\mathscr{G}}. \tag{29}$$

The root mean squared error (RMSE), or L_2-norm, is defined as:

$$D_{\text{RMSE}} = \|\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}\|, \tag{30}$$

with $\boldsymbol{\alpha} \in \Delta^{n-1}$. Last, the Kullback–Leibler (KL) divergence is calculated as:

$$D_{\text{KL}} = \sum_{i=1}^{n} \alpha \log \frac{\alpha_i}{\hat{\alpha}_i}. \tag{31}$$



(a) **(b)**

Figure 8. Visualisation of the Gram matrices . (**a**) Gram matrix of kernel k on the process settings $\boldsymbol{X}$. (**b**) Gram matrix of kernel l on the grid.

6. Results and Discussion

In Figure A4, results for four different experiments are visualized: the measured distribution and the predicted distribution using LOOCV. For the two figures on the left, the calibrated distributions using a PBM from Van Hauwermeiren et al. [1] are plotted as well. Calibrated distributions using the PBM are not available for the experiments in the two right figures, so only predicted distributions using KME are plotted there. More figures with results can be found in Appendix A. These figures show that a good prediction can be achieved over a wide range of distribution shapes (monomodal, monomodal with high skewness, and bimodal) without any assumption of the nature of the true underlying

distribution. Some peculiarities occur in the predictions: in the first and last bins, the KME model always predicts non-zero values. This might be due to the formulation of the ridge-regression-like problem or incorrect retrieval of the distribution in the pre-imaging problem. A switch to a lasso problem, as described by Grünewälder et al. [33], could potentially alleviate this issue. Alternatively, the entropic regularization in Equation (27) might not be chosen in an optimal fashion. A closer look into experiments 9, 12, and 14 shows that the KME predicts the location of the peaks and the skewness of the distributions accurately. Experiments 9 and 14 have lower values of MMD, RMSE, and KL compared to the previous work using PBMs. Considering experiment 20, the model correctly ignores the measurement error of slight bimodal distribution. Overall, when ignoring the peculiarities in the first and last bins, the model attains better results than previous work using population balance models (PBMs) [1], while at the same time lowering the computational cost of calibration and validation and the number of required parameters to describe the model. The main difference in the resulting distributions between KME and PBM is the amount of separation between the modes in bimodal distributions. In other words, the calibrated distributions using PBM have more separated peaks, which results in a recurrent underprediction between the modes in the middle of the distribution and overprediction at the modes. In terms of objective function value, the PBM work quantified the distance between simulation and measurement using the root mean squared error. In Van Hauwermeiren et al. [1], the average RMSE value is 0.0803. In this work, the average RMSE value is 0.0876. For the MMD, the average values for the KME are higher than the PBM approach. According to the KL distance, the KME approach gave better results than the PBM work. Note that in the aforementioned work, the published results are calibration, whereas, in this work, one extra level of complexity is added: predictions of unseen data. Thus, the results presented in this work using KME achieve similar goodness-of-fit while trying to solve a more complex problem. An overview of all goodness-of-fit values can be found in Table 2. It should be noted that, to the authors' best knowledge, this work is the first one to compare PBM with KME.

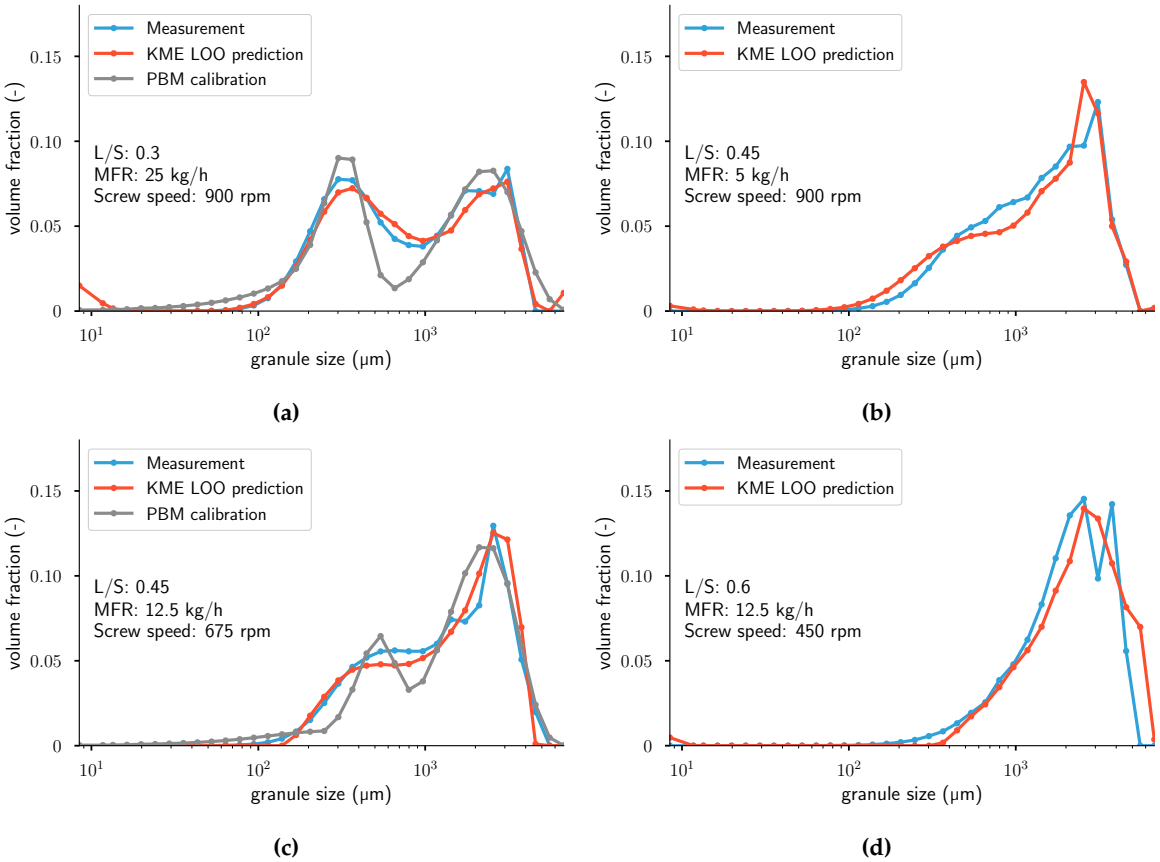


Figure 9. Measurements and leave-one-out cross-validation (LOOCV) predictions of $\mathbb{P}_{Y|X}$. For the two figures on the left, the calibrated distributions using a PBM model from Van Hauwermeiren et al. [1] are plotted as well. From top left to bottom right: (**a**) Experiment 9. (**b**) Experiment 12. (**c**) Experiment 14. (**d**) Experiment 20. Note that the population balance model (PBM) calibration was not performed for experiments (b) and (d). KME: kernel mean embedding; MFR: mass flow rate.

Table 2. Overview of the quality of the LOOCV KME prediction and the calibration of the previous PBM model [1] for each experiment expressed in three distance functions: maximum mean discrepancy (MMD), root mean square error (RMSE), and Kullback–Leibler divergence (KL). At the bottom, the mean of each column is added. The columns of the PBM are missing some values because the calibration could not be performed for those experiments due to missing data in the wetting zone. For more information, see Van Hauwermeiren et al. [1].

Experiment	MMD KME	MMD PBM	RMSE KME	RMSE PBM	KL KME	KL PBM
1	8.10×10^{-4}	3.28×10^{-3}	5.05×10^{-2}	5.92×10^{-2}	3.82×10^{-1}	3.02×10^{-1}
2	4.73×10^{-3}		6.22×10^{-2}		5.44×10^{-1}	
3	9.59×10^{-3}	1.57×10^{-3}	1.00×10^{-1}	7.49×10^{-2}	1.84×10^{0}	7.61×10^{-1}
4	5.22×10^{-3}		5.06×10^{-2}		5.99×10^{-1}	
5	2.28×10^{-3}		6.10×10^{-2}		3.10×10^{-2}	
6	1.94×10^{-4}		2.48×10^{-2}		1.36×10^{-1}	
7	7.04×10^{-3}	8.05×10^{-4}	9.56×10^{-2}	7.41×10^{-2}	1.93×10^{0}	1.80×10^{0}
8	3.10×10^{-3}		4.42×10^{-2}		3.30×10^{-2}	
9	7.58×10^{-4}	4.26×10^{-3}	3.16×10^{-2}	6.49×10^{-2}	5.87×10^{-1}	1.04×10^{0}
10	2.10×10^{-3}		7.78×10^{-2}		2.30×10^{0}	
11	4.39×10^{-3}		5.78×10^{-2}		5.15×10^{-1}	
12	1.16×10^{-3}		5.04×10^{-2}		1.06×10^{-1}	
13	4.02×10^{-2}		1.16×10^{-1}		4.72×10^{-1}	
14	8.60×10^{-4}	2.71×10^{-3}	4.62×10^{-2}	6.50×10^{-2}	3.60×10^{-2}	2.28×10^{-1}
15	5.46×10^{-4}		6.22×10^{-2}		4.22×10^{-1}	
16	1.15×10^{-2}		7.74×10^{-2}		4.76×10^{-1}	
17	5.27×10^{-2}		1.55×10^{-1}		3.53×10^{0}	
18	3.73×10^{-2}	7.26×10^{-4}	1.63×10^{-1}	1.10×10^{-1}	9.09×10^{-1}	1.92×10^{0}
19	1.39×10^{-3}	1.07×10^{-3}	6.00×10^{-2}	5.76×10^{-2}	9.35×10^{-1}	7.98×10^{-1}
20	4.90×10^{-3}		9.76×10^{-2}		2.34×10^{0}	
21	6.51×10^{-3}		8.01×10^{-2}		5.39×10^{-1}	
22	4.99×10^{-2}		1.55×10^{-1}		4.45×10^{-1}	
23	4.58×10^{-2}	1.52×10^{-3}	1.38×10^{-1}	6.83×10^{-2}	9.43×10^{-1}	7.34×10^{-1}
24	6.31×10^{-3}		9.74×10^{-2}		9.75×10^{-2}	
25	5.53×10^{-2}	3.52×10^{-3}	3.01×10^{-1}	1.48×10^{-1}	9.82×10^{-1}	3.21×10^{0}
26	2.61×10^{-3}		3.70×10^{-2}		3.12×10^{-2}	
27	8.45×10^{-3}		1.16×10^{-1}		7.64×10^{-2}	
28	8.80×10^{-3}		6.52×10^{-2}		5.45×10^{-2}	
29	4.31×10^{-3}		6.64×10^{-2}		9.46×10^{-2}	
mean	1.31×10^{-2}	2.16×10^{-3}	8.76×10^{-2}	8.03×10^{-2}	7.38×10^{-1}	1.20×10^{0}

7. Conclusions

The KME of distributions is an interesting data-driven framework to learn relations between a certain input space (in this application, TSWG process settings) and measured distributions. It can be written into a form that allows the description of the problem as a kernel ridge regression problem. Using this framework, kernel mean embeddings of distributions can be predicted given certain inputs. With the kernel pre-image problem, the prediction can be translated from the high-dimensional Hilbert space into its original space. This allows the interpretation and evaluation of the framework in its original space. The benefits of using KME are fast calculation (a couple of seconds for the given problem), analytical short-cuts for LOOCV, high-quality predictions for a wide variety of distribution shapes without making any assumption about those distributions, and a small number of parameters. The model only has five hyperparameters, for which only the regularization parameter λ was estimated.

This work shows an intuitive data-driven approach for which the whole workflow can be written in less than 30 lines of code (see Appendix B). This compactness combined with the fact that only one parameter needs to be estimated makes it an attractive choice for practitioners with limited programming knowledge.

The whole calculation of the LOOCV KME and pre-image problem takes only a couple of seconds. This is in stark contrast with the PBM calculation from previous work [1], where a single calculation of a PBM already takes a couple of seconds. Performing the whole parameter estimation to yield the results that were also shown in the previous section takes orders of magnitude more time than our data-driven approach.

In conclusion, the proposed approach to predict PSDs in TSWG is fast, does not make any assumptions about the shape of the data, and most importantly, yields high-quality cross-validated results.

8. Prospects

Further improvements of this work could include an extension to learn relationships from distribution to distribution. In this application field, this could be applied to assessing the effect of a change in pre-blend composition on the resulting PSD at the end of the granulator. Further, instead of working with one unit operation, the whole ConsiGma^{TM}-25 production line could be studied. The whole transformation of distributions in the unit operations (feeder, blender, granulator, dryer, mill, and tablet press) could be mapped.

The data for this work was gathered using the off-line measurement device QICPIC. It might be interesting to investigate if the model can be trained with a similar predictive power using an in-line particle size measurement device. In that way, the time-consuming preprocessing steps could be bypassed and the results could be gathered more swiftly. If the same experimental design was used, all the data could be gathered in a matter of hours. However, in-line measurement is more prone to noise and has a lower resolution. The effect of this on the model predictions needs to be investigated.

This method is here applied to particle size distributions, but could be extended to other types of distribution-like data. This work focused on a distribution of particle size, but for instance, the moisture distribution in a collection of granules or the hardness of a representative set of final tablets are also possible applications. One other example seems obvious: the prediction of mixtures. In this sense, we could answer questions like "what is the behavior of a mixture of powders starting from the attributes of its components?" This is a hot topic in pharmaceutical manufacturing, as generating an adequate mixture that has the desired properties is mostly done using expert knowledge. The method described in this work could help in creating a model-based design for mixtures. To our best knowledge, we see no hurdles in applying the same methodology to other data-driven problems with distributed data.

Author Contributions: Data curation: D.V.H.; validation: D.V.H.; writing—original draft, D.V.H.; conceptualization, M.S.; methodology, M.S.; software, D.V.H. and M.S.; writing—review and editing, M.S.; supervision, T.D.B. and I.N. All authors have read and agreed to the published version of the manuscript.

Funding: M.S. is supported by the Research Foundation - Flanders (FWO17/PDO/067).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

doe	design of experiments
iid	independent and identically distributed
KME	kernel mean embedding
MFR	mass flow rate
KL	Kullback–Leibler divergence
LOOCV	leave-one-out cross-validation
L/S	liquid-to-solid ratio
MMD	maximum mean discrepancy
PSD	particle size distribution
RBF	radial basis function
RKHS	recurrent kernel Hilbert space
RMSE	root mean squared error
TSWG	twin-screw wet granulation

Appendix A. Additional Figures

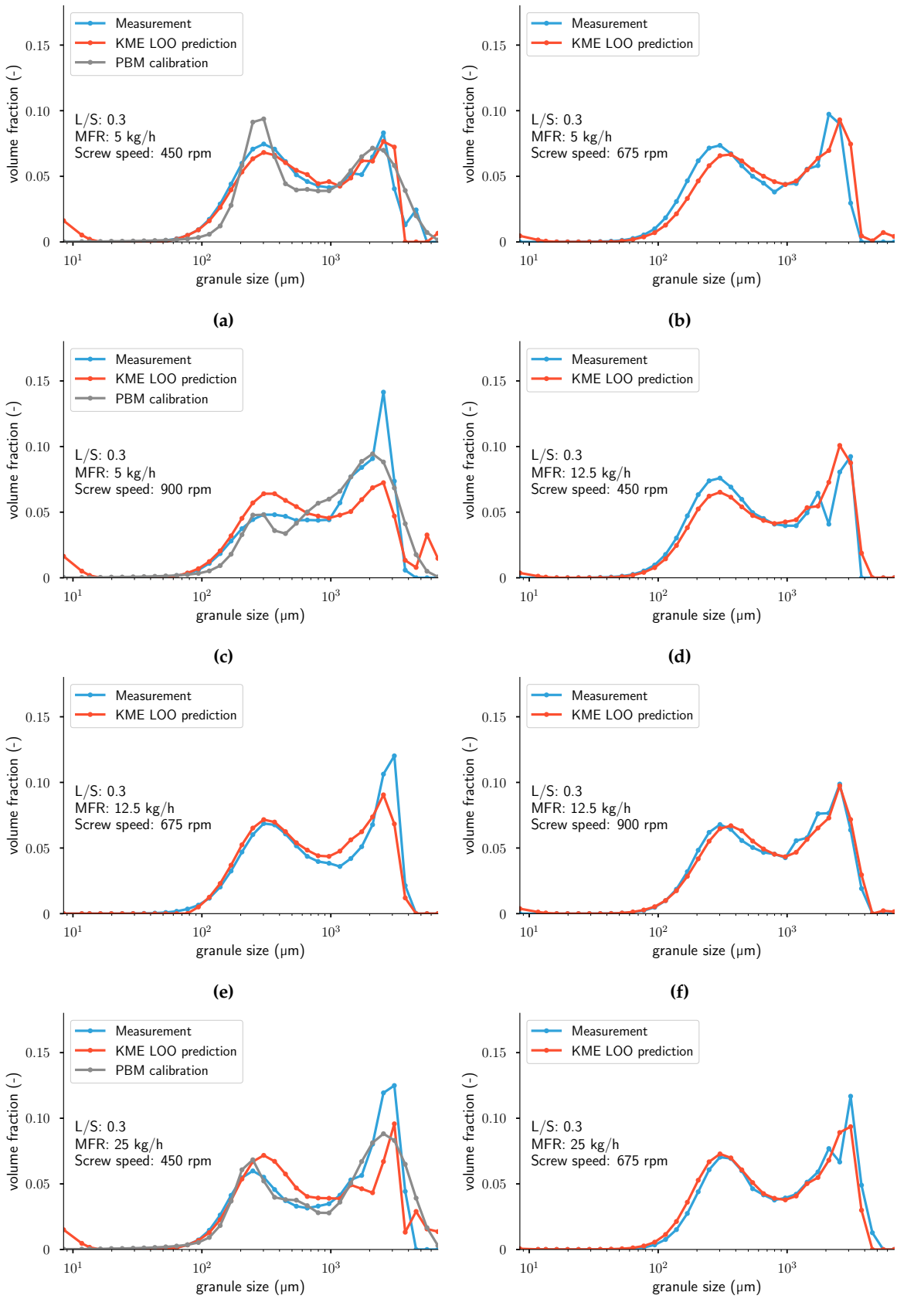


Figure A1. Measurements and leave-one-out cross-validation (LOOCV) predictions of $\mathbb{P}_{Y|X}$. For the two figures on the left, the calibrated distributions using a PBM model from Van Hauwermeiren et al. [1] are plotted as well. From top left to bottom right: (**a**) Experiment 1. (**b**) Experiment 2. (**c**) Experiment 3. (**d**) Experiment 4. (**e**) Experiment 5. (**f**) Experiment 6. (**g**) Experiment 7. (**h**) Experiment 8. Note that the population balance model (PBM) calibration was not performed for experiments (**b**), (**d**), (**e**), (**f**) and (**h**). KME: kernel mean embedding; MFR: mass flow rate.

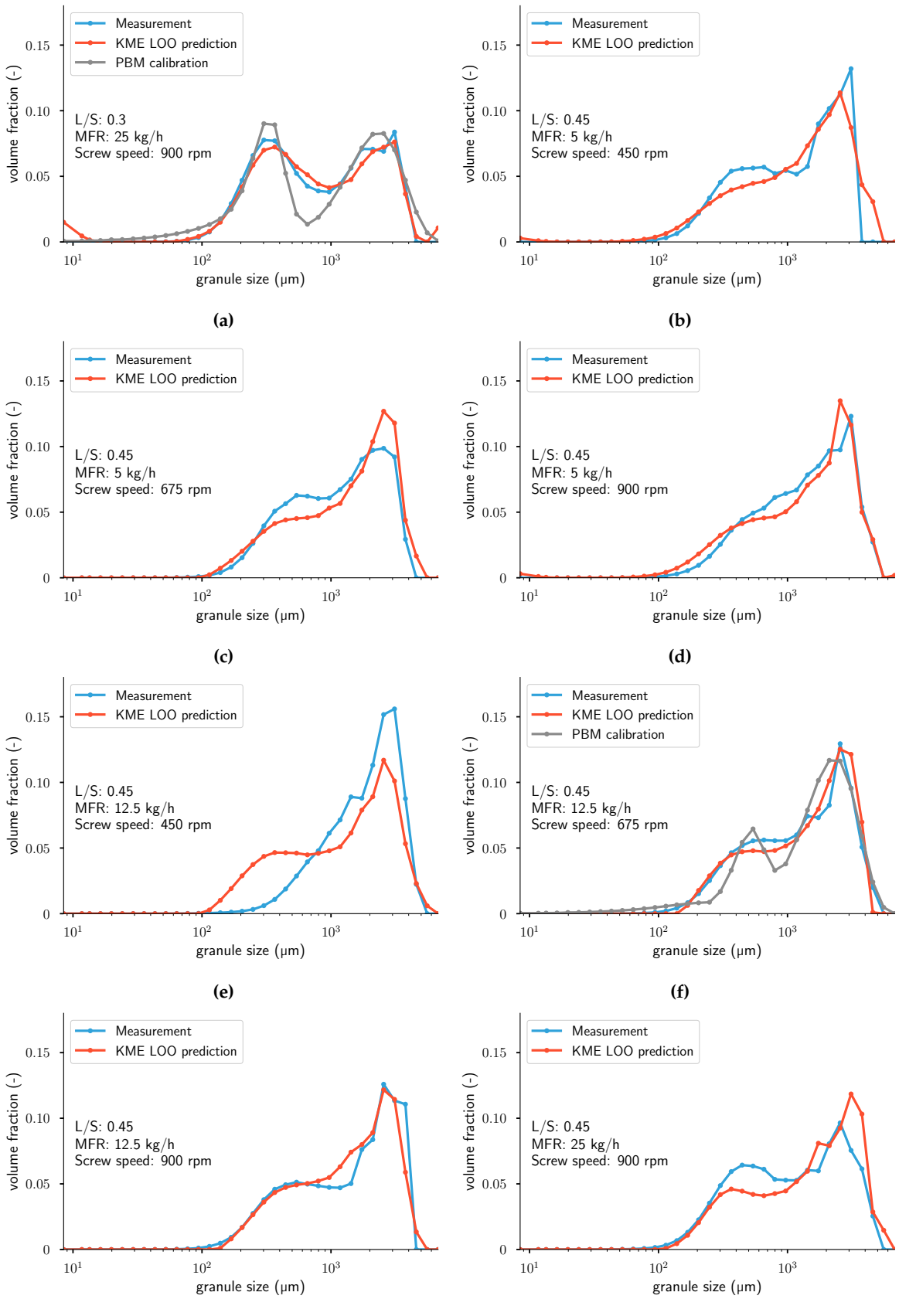


Figure A2. Measurements and leave-one-out cross-validation (LOOCV) predictions of $\mathbb{P}_{Y|X}$. For the two figures on the left, the calibrated distributions using a PBM model from Van Hauwermeiren et al. [1] are plotted as well. From top left to bottom right: (**a**) Experiment 9. (**b**) Experiment 10. (**c**) Experiment 11. (**d**) Experiment 12. (**e**) Experiment 13. (**f**) Experiment 14. (**g**) Experiment 15. (**h**) Experiment 16. Note that the population balance model (PBM) calibration was not performed for experiments (**b**), (**c**), (**d**), (**e**), (**g**) and (**h**). KME: kernel mean embedding; MFR: mass flow rate.

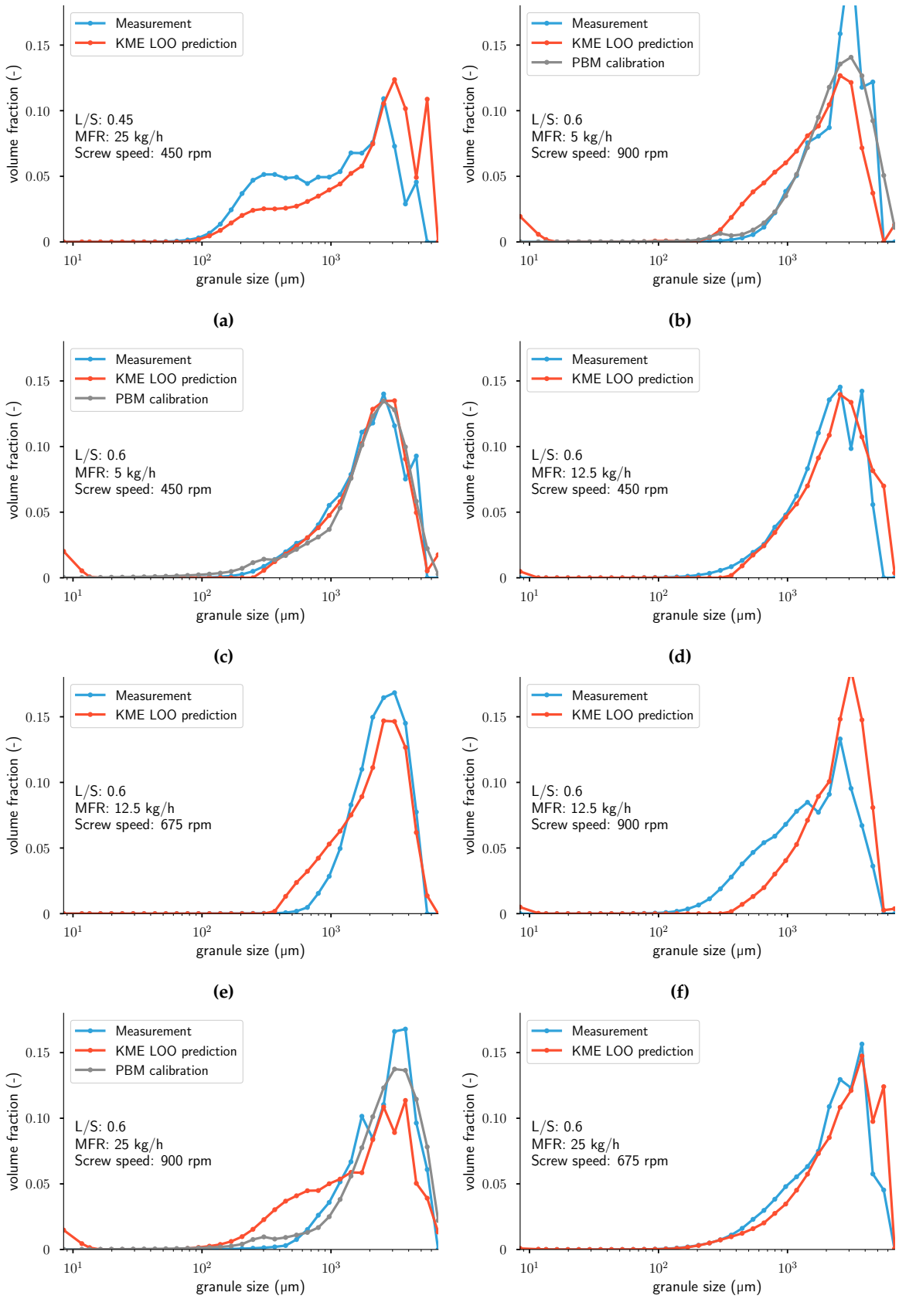


Figure A3. Measurements and leave-one-out cross-validation (LOOCV) predictions of $\mathbb{P}_{Y|X}$. For the two figures on the left, the calibrated distributions using a PBM model from Van Hauwermeiren et al. [1] are plotted as well. From top left to bottom right: (**a**) Experiment 17. (**b**) Experiment 18. (**c**) Experiment 19. (**d**) Experiment 20. (**e**) Experiment 21. (**f**) Experiment 22. (**g**) Experiment 23. (**h**) Experiment 24. Note that the population balance model (PBM) calibration was not performed for experiments (**a**), (**d**), (**e**), (**f**) and (**h**). KME: kernel mean embedding; MFR: mass flow rate.

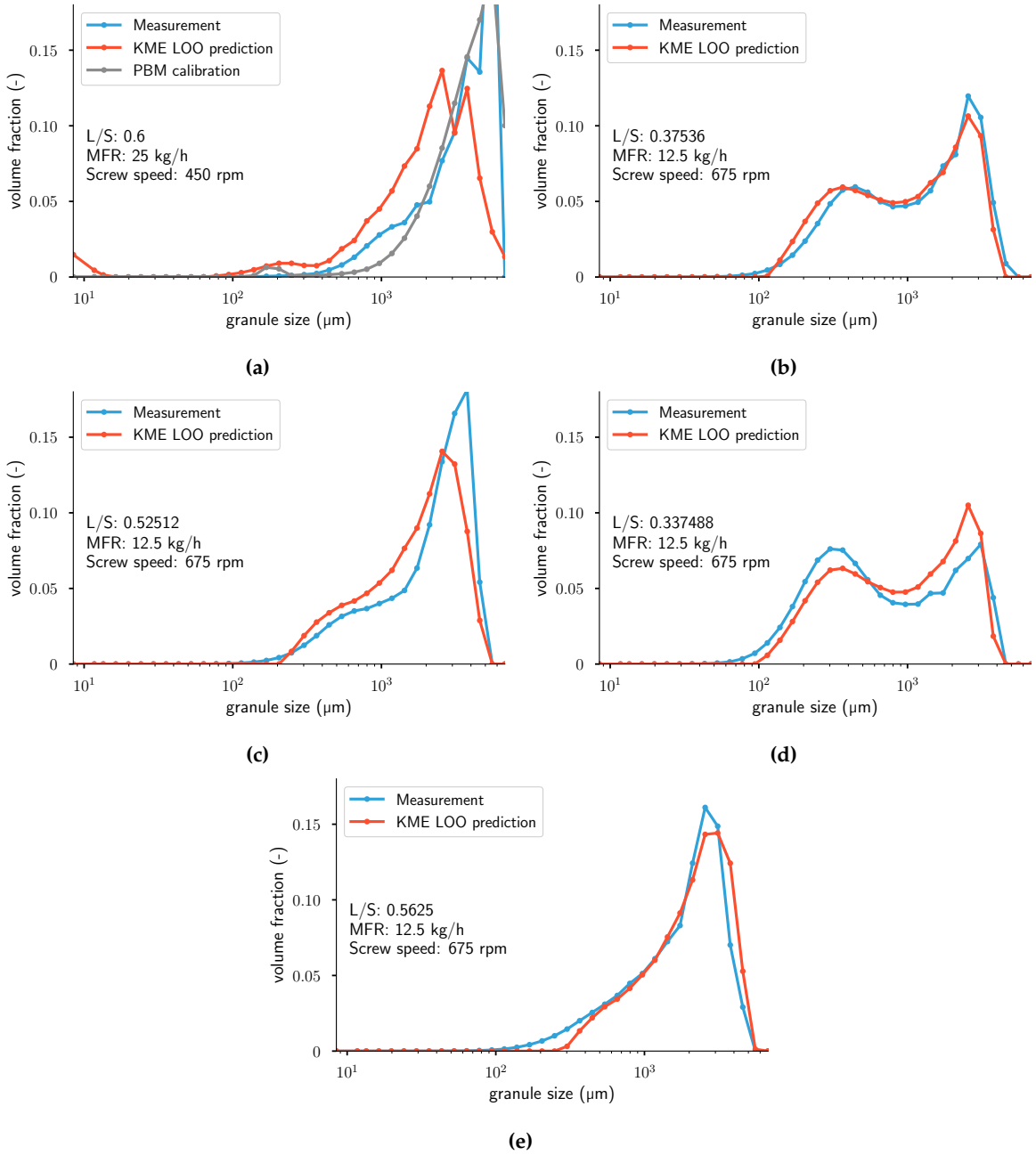


Figure A4. Measurements and leave-one-out cross-validation (LOOCV) predictions of $\mathbb{P}_{Y|X}$. For the two figures on the left, the calibrated distributions using a PBM model from Van Hauwermeiren et al. [1] are plotted as well. From top left to bottom right: (**a**) Experiment 25. (**b**) Experiment 26. (**c**) Experiment 27. (**d**) Experiment 28. (**e**) Experiment 29. Note that the population balance model (PBM) calibration was not performed for experiments (**b**), (**c**), (**d**) and (**e**). KME: kernel mean embedding; MFR: mass flow rate.

Appendix B. Julia Code

```
import MLKernels, LinearAlgebra, Statistics.mean
using JuMP, Gurobi

"""
    calculate_KME(Z, x, A, σ_RBF_grid, bias_grid, σ_RBF_procvars,
                  bias_procvars, λ)

Calculate a ridge regression problem of conditional distributions `A`,
which were measured on the support `x`using Kernel Mean Embedding of the
conditional distributions in matrix . The conditing variable is `Z`.
σ_RBF_grid, σ_RBF_procvars are the hyperparameter for the Gaussian kernel to
calculate the Gram matrix on the input space and the distributions,
respectively.
bias_grid, bias_procvars are the bias that is added to the diagonal of the
Gram matrix.
λ is the regularisation parameter in the ridge regression setting.
"""
function calculate_KME(Z, x, A, σ_RBF_grid, bias_grid, σ_RBF_procvars,
        bias_procvars, λ)
    n_distr, n_classes = size(A)
    # generating the kernel over the grid
    K = MLKernels.kernelmatrix(
        MLKernels.Val(:row),
        MLKernels.RadialBasisKernel(σ_RBF_grid),
        reshape(log10.(x),:,1))
    # adding the bias
    K += bias_grid*I
    # kernel over the process variables
    C = MLKernels.kernelmatrix(
        MLKernels.Val(:row),
        MLKernels.RadialBasisKernel(σ_RBF_procvars),
        Z)
    # adding the bias
    C += bias_procvars*I
    # kernel over the output distributions
    Q = A * K
    Γ =  Q * A'
    Γ_centered = Γ .- mean(Γ, dims=1) .- mean(Γ, dims=2) .+ mean(Γ)
    # building a model
    H = C / (C + λ*I)
    F = H * Q
    F_loo = (I - Diagonal(H)) \ (F - Diagonal(H) * Q)
    # MMD by LOOCV
    mean((Q - F_loo).^2)
    # recovering information from the mean embedding
    predicted_weights = Array{Float64}(undef, (n_distr, n_classes))
    for i in 1:n_distr
        model = JuMP.Model(with_optimizer(Gurobi.Optimizer, OutputFlag=0))
        @variable(model, β[1:n_classes] >= 0.0)
        @constraint(model, sum(β) == 1.0)
        @objective(model, Min, sum(β' * K * β) - 2dot(β, F_loo[i,:]))
        optimize!(model)
        predicted_weights[i,:] = JuMP.value.(β)
    end
    K, C, Γ, predicted_weights, Γ_centered
end
```

References

1. Van Hauwermeiren, D.; Verstraeten, M.; Doshi, P.; am Ende, M.T.; Turnbull, N.; Lee, K.; De Beer, T.; Nopens, I. On the modelling of granule size distributions in twin-screw wet granulation: calibration of a novel compartmental population balance model. *Powder Technol.* **2019**, *341*, 116–125. doi:10.1016/j.powtec.2018.05.025.
2. Verstraeten, M.; Van Hauwermeiren, D.; Lee, K.; Turnbull, N.; Wilsdon, D.; am Ende, M.; Doshi, P.; Vervaet, C.; Brouckaert, D.; Mortier, S.T.; et al. In-depth experimental analysis of pharmaceutical twin-screw wet granulation in view of detailed process understanding. *Int. J. Pharm.* **2017**, *529*, 678–693. doi:10.1016/j.ijpharm.2017.07.045.

3. Gärtner, T. A survey of kernels for structured data. *ACM SIGKDD Explor. Newsl.* **2003**, *5*, 49. doi:10.1145/959242.959248.
4. Hofmann, T.; Schölkopf, B.; Smola, A.J. Kernel methods in machine learning. *Ann. Stat.* **2008**, *36*, 1171–1220, doi:10.1214/009053607000000677.
5. Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; Schölkopf, B. Kernel Mean Embedding of Distributions: A Review and Beyond. *Found. Trends® Mach. Learn.* **2016,** doi:10.1561/2200000060.
6. Schölkopf, B.; Smola, A.J.; Bach, F. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
7. Berlinet, A.; Thomas-Agnan, C. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*; Springer: Boston, MA, USA, 2004. doi:10.1007/978-1-4419-9096-9.
8. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386–408. doi:10.1037/h0042519.
9. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. doi:10.1007/BF00994018.
10. Pearson, K. On lines and planes of closest fit to systems of points in space. *London Edinburgh Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. doi:10.1080/14786440109462720.
11. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **1933**, *24*, 417–441. doi:10.1037/h0071325.
12. Mercer, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **1909**, *83*, 69–70. doi:10.1098/rspa.1909.0075.
13. Aronszajn, N. Theory of reproducing kernels. *Trans. Am. Math. Soc.* **1950**, *68*, 337–337. doi:10.1090/S0002-9947-1950-0051437-7.
14. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory—COLT '92, Pittsburgh, PA, USA, 27–29 July 1992; ACM Press: New York, NY, USA, 1992; pp. 144–152. doi:10.1145/130385.130401.
15. Vapnik, V. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: Berlin, Germany, 2000; p. 314.
16. Genton, M.G. Classes of kernels for machine learning: A statistics perspective. *J. Mach. Learn. Res.* **2002**, *2*, 299–312.
17. Bochner, S. Monotone funktionen, stieltjessche integrale und harmonische analyse. *Math. Ann.* **1933**, *108*, 378–410. doi:10.1007/BF01452844.
18. Smola, A.; Gretton, A.; Song, L.; Schölkopf, B. A Hilbert space embedding for distributions. In Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT), Sendai, Japan, 1–4 October 2007; pp. 13–31.
19. Fukumizu, K.; Gretton, A.; Sun, X.; Schölkopf, B. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20: 21st Annual Conference on Neural Information Processing Systems 2007*; Biologische Kybernetik, Curran: Red Hook, NY, USA, 2008; pp. 489–496.
20. Sriperumbudur, B.K.; Gretton, A.; Fukumizu, K.; Schölkopf, B.; Lanckriet, G.R.G. Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.* **2010**, *11*, 1517–1561.
21. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A.J. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
22. Schölkopf, B.; Muandet, K.; Fukumizu, K.; Harmeling, S.; Peters, J. Computing functions of random variables via reproducing kernel Hilbert space representations. *Stat. Comput.* **2015**, *25*, 755–766. doi:10.1007/s11222-015-9558-5.
23. Fukumizu, K.; Bach, F.R.; Jordan, M.I. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *J. Mach. Learn. Res.* **2004**, *5*, 73–99.
24. Sriperumbudur, B.K.; Gretton, A.; Fukumizu, K.; Lanckriet, G.; Schölkopf, B. Injective Hilbert space embeddings of robability measures. In Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008), Helsinki, Finland, 9–12 July 2008; Biologische Kybernetik, Omnipress: Madison, WI, USA, 2008; pp. 111–122.
25. Sriperumbudur, B.K.; Fukumizu, K.; Gretton, A.; Schölkopf, B.; Lanckriet, G.R.G. On the empirical estimation of integral probability metrics. *Electron. J. Stat.* **2012**, *6*, 1550–1599. doi:10.1214/12-EJS722.
26. Song, L.; Fukumizu, K.; Gretton, A. Kernel embeddings of conditional distributions: a unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Process. Mag.* **2013**, *30*, 98–111. doi:10.1109/MSP.2013.2252713.

27. Poczos, B.; Xiong, L.; Schneider, J. Nonparametric divergence estimation with applications to machine learning on distributions. *CoRR* **2012**, arXiv:1202.3758
28. Oliva, J.B.; Póczos, B.; Schneider, J.G. Distribution to distribution regression. In Proceedings of the 30th International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013; pp. 1049–1057.
29. Oliva, J.B.; Poczos, B.; Schneider, J. Fast distribution to real regression. In Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS), Reykjavik, Iceland, 22–25 April 2014; pp. 706–714.
30. Song, L.; Huang, J.; Smola, A.; Fukumizu, K. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In Proceedings of the 26th International Conference on Machine Learning (ICML), Montreal, QC, Canada, 14–18 June 2009; pp. 961–968.
31. Fukumizu, K.; Song, L.; Gretton, A. Kernel Bayes' rule: Bayesian inference with positive definite kernels. *J. Mach. Learn. Res.* **2013**, *14*, 3753–3783.
32. Zhang, K.; Peters, J.; Janzing, D.; Schölkopf, B. Kernel-based conditional independence test and application in causal discovery. In Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI), Barcelona, Spain, 14–17 July 2011; pp. 804–813.
33. Grünewälder, S.; Lever, G.; Gretton, A.; Baldassarre, L.; Patterson, S.; Pontil, M. Conditional mean embeddings as regressors. In Proceedings of the 29th International Conference on Machine Learning (ICML), Edinburgh, UK, 26 June–1 July 2012; pp. 1823–1830.
34. Micchelli, C.A.; Pontil, M. On Learning Vector-Valued Functions. *Neural Comput.* **2005**, *17*, 177–204. doi:10.1162/0899766052530802.
35. Wahba, G. *Spline Models for Observational Data*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1990.
36. Kwok, J.T.Y.; Tsang, I.W.H. The pre-image problem in kernel methods. *IEEE Trans. Neural Networks* **2004**, *15*, 1517–1525. doi:10.1109/TNN.2004.837781.
37. Song, L.; Zhang, X.; Smola, A.; Gretton, A.; Schölkopf, B. *Tailoring Density Estimation via Reproducing Kernel Moment Matching*; Proceeding of 25th Annual International Conference on Machine Learning, Helsinki, Finland, 5–9 July, 2008; Association for Computing Machinery: New York, NY, United States, 2008; pp. 992–999. doi:10.1145/1390156.1390281.
38. Kanagawa, M.; Fukumizu, K. Recovering distributions from Gaussian RKHS embeddings. *Aistats* **2014**, *33*, 457–465.
39. Gretton, A.; Herbrich, R.; Smola, A.J.; Bousquet, O.; Schölkopf, B. Kernel methods for measuring independence. *J. Mach. Learn. Res.* **2005**, *6*, 2075–2129.