Journal of Artificial Intelligence Research 79 (2024) 515-567 Submitted 06/2023; published 02/2024

On Mitigating the Utility-Loss in Differentially Private Learning: A New Perspective by a Geometrically Inspired Kernel Approach

Mohit Kumar MOHIT.KUMAR@UNI-ROSTOCK.DE
Faculty of Computer Science and Electrical Engineering
University of Rostock, Germany
and
Software Competence Center Hagenberg GmbH
A-4232 Hagenberg, Austria

Bernhard A. Moser BERNHARD.MOSER@SCCH.AT
Institute of Signal Processing
Johannes Kepler University Linz, Austria
and
Software Competence Center Hagenberg GmbH
A-4232 Hagenberg, Austria

Lukas Fischer LUKAS.FISCHER@SCCH.AT
Software Competence Center Hagenberg GmbH
A-4232 Hagenberg, Austria

Abstract

Privacy-utility tradeoff remains as one of the fundamental issues of differentially private machine learning. This paper introduces a geometrically inspired kernel-based approach to mitigate the accuracy-loss issue in classification. In this approach, a representation of the affine hull of given data points is learned in Reproducing Kernel Hilbert Spaces (RKHS). This leads to a novel distance measure that hides privacy-sensitive information about individual data points and improves the privacy-utility tradeoff via significantly reducing the risk of membership inference attacks. The effectiveness of the approach is demonstrated through experiments on MNIST dataset, Freiburg groceries dataset, and a real biomedical dataset. It is verified that the approach remains computationally practical. The application of the approach to federated learning is considered and it is observed that the accuracy-loss due to data being distributed is either marginal or not significantly high.

1. Introduction

Privacy-preserving machine learning is the central topic of this study. Differential privacy (Dwork & Roth, 2014) is a standard framework to quantify the degree to which the data privacy of each individual in the dataset is preserved while releasing the output of any statistical analysis algorithm. Differential privacy, being a property of an algorithm's data access mechanism, automatically provides protection against arbitrary privacy-leakage risks. The goal of protecting sensitive information (that is embedded in training data) from any leakage through machine learning models has been addressed within the framework of differential privacy (Abadi et al., 2016; Phan et al., 2016). The classical approach for designing differentially private algorithms is *output perturbation*, where the idea is to perturb the

function output via adding noise calibrated to the global *sensitivity* of the function (Dwork et al., 2006). A common form of output perturbation is the *Gaussian mechanism*, where Gaussian noise calibrated to the $L2$ sensitivity is added. Differential privacy has been defined for functions and functional data (Hall et al., 2013). Specifically for functions in RKHS generated by the covariance kernel of the Gaussian process, the correct noise level is established by the sensitivity of the function in the RKHS norm (Hall et al., 2013). The iterative nature of machine learning algorithms causes a high cumulative privacy loss and thus a high amount of noise need to be added to compensate for the privacy loss. A *moments accountant* method (Abadi et al., 2016), based on the properties of a *privacy loss* random variable, has been suggested to keep track of the privacy loss incurred by successive iterations for composition analysis. The moments accountant method can be combined with the use of privacy amplification effect of subsampling to deal with the iterative algorithms (Park et al., 2020).

An obvious effect of adding noise into an algorithm for preserving differential privacy is the loss in algorithm's accuracy. As differential privacy remains immune to any post-processing of released output, the output data can be denoised using statistical estimation theory (Balle & Wang, 2018). It is not surprising that efforts have been made to optimize the privacy-accuracy tradeoff (Geng et al., 2018; Balle & Wang, 2018; Ghosh et al., 2012; Gupte & Sundararajan, 2010; Geng & Viswanath, 2016a; Geng et al., 2015; Geng & Viswanath, 2016b). Previously, the studies (Kumar et al., 2019, 2021) have derived the probability density function of noise that minimizes the expected noise magnitude together with satisfying the sufficient conditions for $(\epsilon, \delta)-$differential privacy. Given N number of $p-$variate data points (represented by a matrix $Y \in \mathbb{R}^{N\times p}$), any computational algorithm operating on the data matrix Y can be represented by a mapping, $alg : \mathbb{R}^{N\times p} \to Range(alg)$. The input perturbation method achieves the $(\epsilon, \delta)-$differential privacy of alg via adding a random noise matrix $V \in \mathbb{R}^{N\times p}$ to Y such that the following inequality holds good:

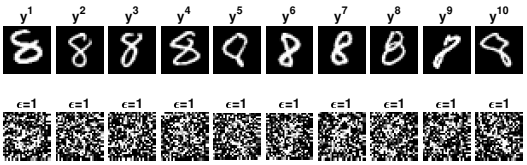
$$Pr\{alg(Y + V) \in \mathcal{O}\} \leq \exp(\epsilon) Pr\{alg(Y' + V) \in \mathcal{O}\} + \delta \tag{1}$$

for any measurable set $\mathcal{O} \subseteq \{alg(Y + \mathrm{V}) \mid Y \in \mathbb{R}^{N\times p}, \mathrm{V} \in \mathbb{R}^{N\times p}\}$ and for *neighboring* matrices pair (Y, Y'). Previously, the noise distribution (from which each element of noise matrix V is independently sampled), that achieves differential privacy inequality (1) with the minimum possible noise magnitude, has been derived (Kumar et al., 2019) using an entropy based approach. The optimal expected noise magnitude is given as (Kumar et al., 2019):

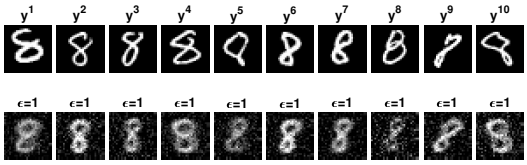
$$E_{f^*_{\mathrm{v}^i_j}}[|v|] = (1 - \delta)\frac{d}{\epsilon}, \tag{2}$$

where $d \in \mathbb{R}_+$ is a scalar defining the adjacency between Y and Y', and v^i_j is the $(i, j)-$th element of noise matrix V with its probability density function as $f_{\mathrm{v}^i_j}(v)$. It follows from (2) that despite an optimization, a low value of privacy-loss bound ϵ requires a large amount of noise leading to a considerable loss in the accuracy of a subsequent machine learning algorithm operating on the noise added data.

To mitigate the effect of noise, the flexibility of defining computational algorithm alg in (1) can be leveraged without compromising on the privacy-loss bound ϵ. Specifically, a



(a) original data samples and corresponding differentially private approximations using optimal noise adding mechanism ($\epsilon = 1$)



(b) original data samples and corresponding differentially private fabricated data samples ($\epsilon = 1$)

Figure 1: An example of the data fabrication by means of a geometric model ensuring the geometric modeling error of fabricated data samples not to exceed that of original data samples without compromising on the value of privacy-loss bound ϵ.

model of the geometric structure induced by noise added data points can be integrated in the definition of *alg* for a *smoothing*. The *alg* can be defined as a composition of a smoothing and the machine learning algorithm:

$$alg := machine\_learning \circ smoothing. \tag{3}$$

Here, the *smoothing* is based on a model (that represents the geometric structure induced by the noise added data points) ensuring that the smoothing leads to the fabrication of new data points which are not only differentially private but also their geometric modeling error does not exceed that of original data points. Fig. 1 provides an example of the data fabrication by means of such a geometric model. The central problem of this study is stated in the following:

Problem 1 (Central Problem)**.** *To mitigate the accuracy-loss issue of differential privacy, the post-processing property of differential privacy is leveraged for fabricating new data samples by means of a geometric model ensuring the geometric modeling error of fabricated data samples not to be larger than that of original data samples while simultaneously achieving the privacy-loss bound.*

Remark 1 (Motivation)**.** *To our best knowledge, the state-of-the-art does not address Problem 1. It requires an approach to learn the representation of geometric structure induced by a finite set of data points. Encouraged by the fact that kernel-based solutions can be computed analytically and analyzed using a broad range of mathematical techniques, the approach opted in this study to address Problem 1 is of learning in Reproducing Kernel Hilbert Spaces (RKHS) the representation of data points to design a geometrically inspired model such that the model output range defines a bounded geometric structure in the affine hull of given data samples.*

Kernels have been widely used in machine learning (Ghojogh et al., 2021; Hofmann et al., 2008) and can be scaled up for their applicability in large scale scenarios (Rudi et al., 2017). Not only the parallels between the properties of deep neural networks and kernel methods have been established (Belkin et al., 2018), but also deep kernel machines have been introduced (Wilson et al., 2016; Nikhitha et al., 2021). Kernel autoencoders are effective models

for representation learning. The kernel formulation of an autoencoder has been considered in (Gholami & Hajisami, 2016) from a hashing perspective. A deep autoencoder, that aligns the latent code with a user-defined kernel matrix to learn similarity-preserving data representations, has been suggested (Kampffmeyer et al., 2018). Further, a kernel autoencoder based on the composition of mappings from vector-valued reproducing kernel Hilbert spaces has been studied (Laforgue et al., 2019). Recently, a fuzzy theoretic approach to kernel based wide and conditionally deep autoencoders has been introduced (Kumar & Freudenthaler, 2020; Kumar et al., 2021; Zhang et al., 2022; Kumar et al., 2021; Zhang et al., 2023; Kumar et al., 2021b, 2021a, 2023), wherein analytical solutions are derived for the learning of models using variational optimization technique. This approach has been further extended to privacy-preserving learning under a differential privacy framework (Kumar, 2023; Kumar et al., 2021; Kumar, Rossbory, Moser, & Freudenthaler, 2020). As an alternative to the SVM, the idea of affine hull large margin classifier has been investigated (Cevikalp et al., 2010). Although kernel methods have been studied (Jain & Thakurta, 2013; Chaudhuri et al., 2011; Zhang et al., 2019) under differential privacy, no previous study has considered geometrically inspired kernel methods to mitigate the accuracy-loss issue of differential privacy. *State of the art lacks geometrically inspired kernel machines for scalable learning solutions that remain accurate even after providing differential privacy guarantee.*

This study solves Problem 1 via making the following contributions (C1-C7):

Kernel Affine Hull Machines (C1): For given distinct data points $(y^i)_{i=1,\dots,N}$ in some vector space we study the sets of the affine form

$$\mathcal{L} = \left\{ y = \left(w^1 / \sum_{i=1}^{N} w^i \right) y^1 + \cdots + \left(w^N / \sum_{i=1}^{N} w^i \right) y^N \;\mid\; w^i \in \mathbb{R} \right\}, \tag{4}$$

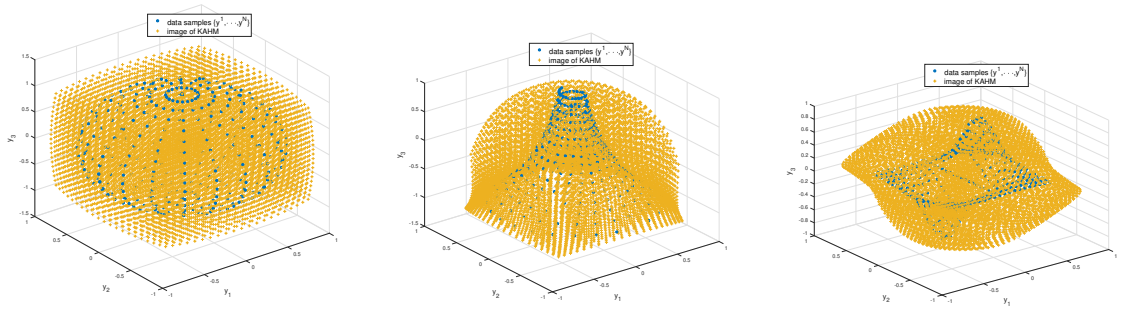
and ask for reasonable conditions on the real-valued scalars $(w^i)_i$ to serve our purpose of representing the geometric structure induced by data points. First of all, in our approach $(w^i)_i$ are considered to be functions in a RKHS. By postulating that indicator functions (specifically, their RKHS approximations) define scalar-valued functions $(w^i)_i$, the set $\mathcal{L}$ actually can be identified by functions defining a subset in RKHS that represents our data points. This way we introduce the concept of Kernel Affine Hull Machine (KAHM) to learn kernel-based representation of multivariate scattered data as in the following:

Let n, p, N be the positive integers and $\mathcal{X} \subset \mathbb{R}^n$ be a region. Let $\mathcal{H}_k(\mathcal{X})$ be the reproducing kernel Hilbert space of functions from $\mathcal{X}$ to $\mathbb{R}$ for a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. For a finite set of ordered pairs $\{(x^i, y^i) \in \mathcal{X} \times \mathbb{R}^p \;\mid\; i \in \{1, \cdots, N\}\}$ such that $\{x^1, \cdots, x^N\}$ are pairwise distinct points, a point y^i can be represented using indicator functions as

$$y^i = \mathbb{1}_{\{x^1\}}(x^i)\, y^1 + \cdots + \mathbb{1}_{\{x^N\}}(x^i)\, y^N, \tag{5}$$

where $\mathbb{1}_{\{x^i\}}$ is the indicator function of the set $\{x^i\}$. We approximate the indicator function $\mathbb{1}_{\{x^i\}}$ through a function in $\mathcal{H}_k(\mathcal{X})$ that fits to the ordered pairs $\{\big(x^j, \mathbb{1}_{\{x^i\}}(x^j)\big) \;\mid\; j \in \{1, \cdots, N\}\}$. The function in RKHS approximating $\mathbb{1}_{\{x^i\}}$ is given as the solution of the following kernel regularized least squares problem:

$$h^i = \arg \min_{f \in \mathcal{H}_k(\mathcal{X})} \left(\sum_{j=1}^{N} \left| \mathbb{1}_{\{x^i\}}(x^j) - f(x^j) \right|^2 + \lambda \left\| f \right\|^2_{\mathcal{H}_k(\mathcal{X})} \right),\ \lambda \in \mathbb{R}_+, \tag{6}$$



(a) first example (b) second example (c) third example

Figure 2: A few examples of 3-dimensional samples $\{y^1, \cdots, y^N\}$ and geometric structures in $\mathrm{aff}(\{y^1, \cdots, y^N\})$ defined by the images of corresponding KAHMs.

where $\|f\|_{\mathcal{H}_k(\mathcal{X})} := \sqrt{\langle f, f \rangle_{\mathcal{H}_k(\mathcal{X})}}$ is the norm induced by the inner product on $\mathcal{H}_k(\mathcal{X})$. The fact that h^i is an approximation of $\mathbb{1}_{\{x^i\}}$ (i.e. the value $h^i(x)$ represents kernel-smoothed "membership" of x to the set $\{x^i\}$) allows introducing a model based on the affine combination of $y^1, \cdots, y^N$ as in the following:

$$A(x) = \frac{h^1(x)}{\sum_{i=1}^N h^i(x)}\, y^1 + \cdots + \frac{h^N(x)}{\sum_{i=1}^N h^i(x)}\, y^N. \tag{7}$$

Let $\mathrm{aff}(\{y^1, \cdots, y^N\})$ denote the affine hull of $\{y^1, \cdots, y^N\}$. The function $A : \mathcal{X} \to \mathrm{aff}(\{y^1, \cdots, y^N\})$ is referred to as kernel affine hull machine, since it maps a point $x \in \mathcal{X}$ onto the affine hull of $\{y^1, \cdots, y^N\}$ via learning representation of $x^1, \cdots, x^N$ through functions in reproducing kernel Hilbert space. The image of A,

$$A[\mathcal{X}] := \{A(x) \mid x \in \mathcal{X}\} \subset \mathrm{aff}(\{y^1, \cdots, y^N\}), \tag{8}$$

defines a geometric structure in $\mathrm{aff}(\{y^1, \cdots, y^N\})$. Fig. 2 displays a few examples of 3-dimensional samples and geometric structures defined by KAHMs' images.

Regularization Parameter for Kernel Regularized Least Squares (C2): Since indicator functions are approximated via solving a regularized least squares problem, the kernel regularized least squares problem is revisited in a deterministic setting with focus on the determination of regularization parameter. A reasonable choice for regularization parameter is to set it larger than the mean-squared-error on training samples. With this choice, the problem of determining regularization parameter can be reduced to an equivalent problem of finding the unique fixed point of a real-valued positive function. An iterative scheme, together with the mathematical proof of convergence, is provided to find the fixed point and thus to determine the regularization parameter.

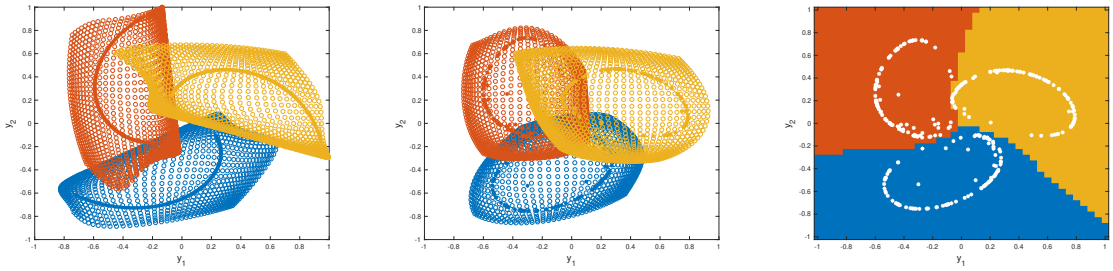
Boundedness of KAHM and Distance Function (C3): The KAHM mapping is a bounded function and thus the image of KAHM defines a bounded region in the affine hull

of data samples. The boundedness of KAHM on data space is proven via deriving upper bounds on the Euclidean norm of KAHM output. The KAHM induces a function on data space, referred to as *distance function*, which is defined on a data point as equal to the distance between that point and its image under KAHM. The distance of an arbitrary point from its image (by the KAHM onto the affine hull of given data samples) is a measure of the distance between that arbitrary point and the given data samples. This is proven via deriving upper bounds on the ratio of these two distances.

KAHM Compositions for Data Representation Learning and Classification (C4): The KAHM could serve as the building block for deep models. A nested composition of KAHMs, referred to as *Conditionally Deep Kernel Affine Hull Machine*, is considered for data representation learning. The conditionally deep KAHM discovers layers of increasingly abstract data representation with lowest-level data features being modeled by first layer and the highest-level by end layer. Further, a parallel composition of conditionally deep KAHMs, referred to as *Wide Conditionally Deep Kernel Affine Hull Machine*, is considered to efficiently learn the representation of big data. Similarly to the KAHM, both conditional deep KAHM and wide conditionally deep KAHM induce the distance function with value on a point indicating the distance of the point from data samples. This property of the distance function is leveraged to build a KAHM based classifier via modeling the region of each class through a separate KAHM based composition.

Membership-Inference Score for KAHM Based Classifier (C5): Since the KAHM based classifier assigns a class-label to a data point based on the closeness of the point to the training data samples of that class, there is a possibility of an inference of the membership of a data point to the set of training data samples. To evaluate the potential of KAHM induced distance function in inferring the membership of a data point to the training dataset, a score, referred to as *membership-inference score*, is defined for evaluating the risk of *membership inference attack*. The membership-inference score is defined as the $L2$ distance between density of probability distribution on values of the distance function at training data points and the density of probability distribution on distance function values at test data points.

Differentially Private Data Fabrication for Classification (C6): To ensure that KAHM based classifier keeps the privacy of training data protected, an optimal differentially private noise adding mechanism (Kumar et al., 2019) is applied on training data samples. The noise added training data samples are smoothed through a transformation such that the error in KAHM modeling of smoothed data is not larger than the error in KAHM modeling of original data. It is shown that the error in KAHM modeling of smoothed data can be reduced to an arbitrary low value. The smoothed data samples, guaranteeing not only the differential privacy but also the geometric modeling error not to be larger than that of original data samples, serve as the *fabricated* data. The fabricated data samples are finally used to build the KAHM based differentially private classifier. The advantage of using fabricated data for classification is that fabricated data leads to a considerable reduction in the risk of membership inference attack with relatively much smaller loss of accuracy. Hence, the accuracy-loss issue of differential privacy is mitigated. Fig. 3 provides an example of differentially private classifier built with a 2-dimensional fabricated dataset with 3 classes.



(a) data samples and images of corresponding KAHMs

(b) differentially private fabricated data samples and images of corresponding KAHMs

(c) decision boundaries determined by KAHM based differentially private classifier with fabricated data

Figure 3: An example of differentially private classifier built with a 2-dimensional fabricated dataset with 3 classes.

Application to Differentially Private Federated Learning (C7): The different KAHMs built independently using different datasets can be combined together using the KAHM induced distance function. This allows introducing a federated learning scheme that combines together the local privacy-preserving KAHM based classifiers to build a global classifier. A significant feature of the scheme is that the evaluation of global classifier requires only locally computed distance measures.

The relation of the current study with previous works is confined to the following three points: 1) The wide and conditionally deep architecture consisting of the composition of kernel based models follows from (Kumar & Freudenthaler, 2020; Kumar et al., 2021, 2021; Zhang et al., 2022; Kumar et al., 2021a), wherein a kernel based variational fuzzy model (motivated by measure theoretic basis (Kumar et al., 2021b)) is used. In contrast, the current study explores geometrically inspired kernel affine hull machines. 2) The input perturbation method (where noise is added to original data to achieve $(\epsilon, \delta)-$differential privacy of any subsequent computational algorithm) was earlier considered in (Kumar et al., 2021, 2020; Kumar, 2023). However, the current study complements the input perturbation method with a transformation to mitigate the accuracy-loss issue of differential privacy. 3) The current study follows the federated learning architecture of (Kumar et al., 2021, 2020, 2023) with the difference that instead of fuzzy attributes, the KAHM induced distance measures are applied to aggregate the distributed local models for federated learning.

The significance and novelties of the contributions have been highlighted in Table 1 and Table 2 respectively.

The paper is organized into the following sections. The mathematical notation used throughout the paper is provided in Section 2. Section 3 introduces the concept of KAHM. KAHM based wide and deep models are presented in Section 4 for data representation learning. Differentially private classification application is considered in Section 5 followed by experimentation in Section 6. Finally, the concluding remarks are presented in Section 7.

Table 1: The significance of the contributions.

	Significance
C1	Representations learning in RKHS for defining a geometric structure in the affine hull of data samples
C2	Determination of the regularization parameter for kernel regularized least squares
C3	Because of the boundedness of KAHM, the distance of an arbitrary point from its KAHM image is a measure of the distance between that arbitrary point and the given data samples
C4	KAHM compositions learn geometrically inspired representations at varying abstraction level facilitating classification via modeling the region of each class through a separate composition
C5	Evaluation of the risk of membership inference attack on KAHM based classifier
C6	Differentially private data fabrication to mitigate the accuracy-loss issue associated with the differentially private classifier
C7	Application to differentially private federated learning

Table 2: The novelties in the contributions.

	Novelty
C1	The concept of KAHM (Definition 1) is novel.
C2	Determination of the regularization parameter as the unique fixed point of a function (Theorem 1) is novel.
C3	The idea of using bounded geometric structure (Theorem 2) to define a measure of the distance from given data samples (Theorem 3) is novel.
C4	Geometrically inspired representations learning at varying abstraction level and corresponding induced measure of the distance from given data samples (Theorem 4, Theorem 5) is novel.
C5	Quantification of membership inference attack risk as $L2$ distance between density of distance from training data samples and density of distance from test data samples (Eq. (69)) is novel.
C6	Data fabrication via transforming differentially private data samples to reduce geometric modeling error (Definition 13, Theorem 6, Definition 14) is novel.
C7	The feature of the federated learning that the evaluation of global classifier requires only locally computed KAHM induced distance measures (Fig. 11) is novel.

2. Notations

The following notations are introduced:

- Let $N, n, p, M, S, C \in \mathbb{Z}_+$ be the positive integers.
- Let $\mu_{max}(K)$ and $\mu_{min}(K)$ denote the maximum eigenvalue and minimum eigenvalue respectively of a square matrix K.
- Let $\sigma_{max}(K)$ and $\sigma_{min}(K)$ denote the maximum singular value and minimum singular value of a matrix K.
- Let $\text{aff}(\mathbf{Y})$ denote the affine hull of a set $\mathbf{Y} \subset \mathbb{R}^p$.
- For a vector $y \in \mathbb{R}^p$, $\|y\|$ denotes the Euclidean norm of y.

- For a matrix $Y \in \mathbb{R}^{N\times p}$, $\|Y\|_2$ denotes the spectral norm, $\|Y\|_F$ denotes the Frobenius norm, $\|Y\|_1$ denotes the 1-norm, $\|Y\|_{\max}$ denotes the max norm, $(Y)_{i,:}$ denotes the $i-$th row, $(Y)_{:,j}$ denotes the $j-$th column, and $(Y)_{i,j}$ denotes the $(i,j)-$th element of Y.
- Let $\circ$ denote the Hadamard product.
- Let I_N denote the identity matrix of the size N and $\mathbf{1}_N$ denotes the $N \times 1$ vector of ones.
- Let $\mathbb{1}_{\mathbf{Y}}$ denote the indicator function of the set $\mathbf{Y}$.
- Let $\mathcal{X} \subset \mathbb{R}^n$ be a region.
- $K \succ 0$ denotes that a symmetric matrix K is positive definite.
- A Reproducing Kernel Hilbert Space (RKHS), $\mathcal{H}_k(\mathcal{X})$, is a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ on a non-empty set $\mathcal{X}$ with a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{H}$,
  - $k(\cdot, x) \in \mathcal{H}_k(\mathcal{X})$,
  - $\langle f, k(\cdot, x)\rangle_{\mathcal{H}_k(\mathcal{X})} = f(x)$,

  where $\langle\cdot,\cdot\rangle_{\mathcal{H}_k(\mathcal{X})} : \mathcal{H}_k(\mathcal{X}) \times \mathcal{H}_k(\mathcal{X}) \to \mathbb{R}$ is an inner product on $\mathcal{H}_k(\mathcal{X})$.
- Let $\|f\|_{\mathcal{H}_k(\mathcal{X})} := \sqrt{\langle f, f\rangle_{\mathcal{H}_k(\mathcal{X})}}$ denote the norm induced by the inner product on $\mathcal{H}_k(\mathcal{X})$.

3. Kernel Affine Hull Machines

The computation of KAHM requires solving a kernel regularized least squares problem. Therefore the kernel regularized problem is revisited (in Section 3.1) with focus on the determination of regularization parameter (in Section 3.2). The obtained solution is applied (in Section 3.3) to learn data representation in RKHS facilitating the definition of KAHM (in Section 3.4).

3.1 Kernel Regularized Least Squares

Given a training data set $\left\{(x^i, y^i) \in \mathcal{X} \times \mathbb{R}^p \;\middle|\; i \in \{1, \cdots, N\}\right\}$ such that $\{x^1, \cdots, x^N\}$ are pairwise distinct points, consider a positive-definite real-valued kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ on $\mathcal{X}$ with a corresponding RKHS $\mathcal{H}_k(\mathcal{X})$. Assuming that $\mathcal{X} \subset \mathbb{R}^n$, real-valued matrices $X \in \mathbb{R}^{N\times n}$ and $Y \in \mathbb{R}^{N\times p}$ are defined as

$$X = \left[\, x^1 \;\cdots\; x^N \,\right]^T \tag{9}$$

$$Y = \left[\, y^1 \;\cdots\; y^N \,\right]^T. \tag{10}$$

Let $(Y)_{:,j}$ denote the $j-$th column of Y, i.e.,

$$(Y)_{:,j} = \left[\, y_j^1 \;\cdots\; y_j^N \,\right]^T, \tag{11}$$

where $j \in \{1, \cdots, p\}$ and y_j^i is the $j-$th element of $i-$th output sample y^i. The solution of the following regularized least squares problem:

$$f^*_{k,X,(Y)_{:,j},\lambda} = \arg \min_{f \in \mathcal{H}_k(\mathcal{X})} \left(\sum_{i=1}^{N} \left| y_j^i - f(x^i) \right|^2 + \lambda \left\| f \right\|^2_{\mathcal{H}_k(\mathcal{X})} \right), \ \lambda \in \mathbb{R}_+, \tag{12}$$

using the representer theorem (Schölkopf, Herbrich, & Smola, 2001), can be written as

$$f^*_{k,X,(Y)_{:,j},\lambda}(x) = \left[k(x, x^1) \ \cdots \ k(x, x^N) \right] (K_X + \lambda I_N)^{-1} (Y)_{:,j} \tag{13}$$

where I_N is the identity matrix of size N and K_X is $N \times N$ kernel matrix whose $(i,j)-$th entry is given as

$$(K_X)_{ij} = k(x^i, x^j). \tag{14}$$

The regularized least squares problem can be solved for each output dimension resulting in a vector-valued function $\mathbf{f}^*_{k,X,Y,\lambda} : \mathcal{X} \to \mathbb{R}^p$ defined as

$$\mathbf{f}^*_{k,X,Y,\lambda}(x) := \left[f^*_{k,X,(Y)_{:,1},\lambda}(x) \ \cdots \ f^*_{k,X,(Y)_{:,p},\lambda}(x) \right]^T \tag{15}$$

$$= Y^T (K_X + \lambda I_N)^{-1} \left[k(x, x^1) \ \cdots \ k(x, x^N) \right]^T. \tag{16}$$

3.2 Determination of Regularization Parameter

To compute the kernel regularized least squares solution (13), a choice for regularization parameter $\lambda \in \mathbb{R}_+$ need to be made. A possible choice could be of setting λ larger than the mean squared error on training data. The mean squared error on training data, which obviously depends on the choice of regularization parameter λ, is given as

$$e(\lambda) = \frac{1}{pN} \sum_{j=1}^{p} \sum_{i=1}^{N} |y_j^i - f^*_{k,X,(Y)_{:,j},\lambda}(x^i)|^2 \tag{17}$$

$$= \frac{1}{pN} \sum_{j=1}^{p} \| (Y)_{:,j} - K_X (K_X + \lambda I_N)^{-1} (Y)_{:,j} \|^2. \tag{18}$$

We choose λ to be larger than e. That is, there exists a constant $\tau \in \mathbb{R}_+$ such that

$$\lambda = e(\lambda) + \tau, \ \text{i.e.,} \tag{19}$$

$$\lambda = \frac{1}{pN} \sum_{j=1}^{p} \| (Y)_{:,j} - K_X (K_X + \lambda I_N)^{-1} (Y)_{:,j} \|^2 + \tau. \tag{20}$$

Eq. (20) can be solved for λ via applying the following result:

Theorem 1. *Let $\mathcal{R}_{k,X,Y} : \mathbf{R}_+ \times \mathbf{R}_+ \to \mathbf{R}_+$ be a function defined as*

$$\mathcal{R}_{k,X,Y}(e, \tau) := \frac{1}{pN} \sum_{j=1}^{p} \| (Y)_{:,j} - K_X (K_X + (e + \tau) I_N)^{-1} (Y)_{:,j} \|^2. \tag{21}$$

We have followings:

1. *For* $e, \tau \in \mathbf{R}_+$,

$$\mathcal{R}_{k,X,Y}(e,\tau) \in (0, \frac{\|Y\|_F^2}{pN}). \tag{22}$$

2. *For* $e, \tau \in \mathbf{R}_+$,

$$\frac{\mathrm{d}\mathcal{R}_{k,X,Y}(e,\tau)}{\mathrm{d}e} \in (0, \frac{2}{(e+\tau)}\frac{\|Y\|_F^2}{pN}). \tag{23}$$

3. *For a given* $\tau \in \mathbf{R}_+$, $\mathcal{R}_{k,X,Y}(e,\tau)$ *has at least one fixed point in* $(0, \|Y\|_F^2/pN)$.

4. *If we choose*

$$\tau \geq \frac{2}{pN}\|Y\|_F^2, \tag{24}$$

then the iterations

$$e|_{it+1} = \mathcal{R}_{k,X,Y}(e|_{it}, \tau),\ it \in \{0, 1, \cdots\} \tag{25}$$

$$e|_0 \in (0, \frac{\|Y\|_F^2}{pN}) \tag{26}$$

converge to the unique fixed point of $\mathcal{R}_{k,X,Y}(e,\tau)$.

Proof. The proof is provided in Appendix A. □

It follows from Theorem 1 that the iterations (25)-(26) converge to the unique fixed point of $\mathcal{R}_{k,X,Y}(e,\tau)$ for any τ satisfying (24). Let $\hat{e}$ be the unique fixed point corresponding to the minimum possible value of τ satisfying (24) (which is equal to $\frac{2}{pN}\|Y\|_F^2$), i.e.,

$$\hat{e} = \mathcal{R}_{k,X,Y}(\hat{e}, \frac{2}{pN}\|Y\|_F^2). \tag{27}$$

Now, the value of regularization parameter λ satisfying (20) for $\tau = \frac{2}{pN}\|Y\|_F^2$ is given as

$$\lambda^* = \hat{e} + \frac{2}{pN}\|Y\|_F^2. \tag{28}$$

3.3 Learning Representation of Data Points in RKHS

Given a finite number of pairwise distinct points: $X = [\, x^1 \ \cdots \ x^N \,]^T$ with $x^1, \cdots, x^N \in \mathcal{X} \subset \mathbb{R}^n$, a data point x^i can be represented using indicator functions as

$$x^i = \mathbb{1}_{\{x^1\}}(x^i)\, x^1 + \cdots + \mathbb{1}_{\{x^N\}}(x^i)\, x^N, \text{ for any } i \in \{1, \cdots, N\}. \tag{29}$$

For a kernel-based representation of data points, the indicator functions $\mathbb{1}_{\{x^1\}}, \cdots, \mathbb{1}_{\{x^N\}}$ are approximated through functions in RKHS $\mathcal{H}_k(\mathcal{X})$. To approximate $\mathbb{1}_{\{x^i\}}$, a function is

fitted on the ordered pairs $\{\big(x^j, \mathbb{1}_{\{x^i\}}(x^j)\big) \mid j \in \{1, \cdots, N\}\}$ via solving the following kernel regularized least squares problem:

$$h^i_{k,X,\lambda} = \arg \min_{f \in \mathcal{H}_k(\mathcal{X})} \left(\sum_{j=1}^{N} \left| \mathbb{1}_{\{x^i\}}(x^j) - f(x^j) \right|^2 + \lambda \left\| f \right\|^2_{\mathcal{H}_k(\mathcal{X})} \right), \ \lambda \in \mathbb{R}_+. \tag{30}$$

Using the representer theorem (Schölkopf et al., 2001), the solution of (30) is as follows:

$$h^i_{k,X,\lambda}(x) = (I_N)_{i,:} \left(K_X + \lambda I_N\right)^{-1} \left[k(x, x^1) \ \cdots \ k(x, x^N) \right]^T, \tag{31}$$

where K_X is kernel matrix defined as in (14) and $(I_N)_{i,:}$ denotes the $i-$th row of identity matrix of size N. The function $h^i_{k,X,\lambda} : \mathcal{X} \to \mathbb{R}$ is a kernel-smoothed approximation of $\mathbb{1}_{\{x^i\}} : \mathcal{X} \to \{0, 1\}$, and thus $h^i_{k,X,\lambda}(x)$ represents the kernel-smoothed "membership" of x to the set $\{x^i\}$. Given a response variable $y^i \in \mathbb{R}^p$ associated to data point x^i, a regression model based on the affine combination of response variables can be defined as in the following:

$$A(x) = \frac{h^1_{k,X,\lambda}(x)}{\sum_{i=1}^N h^i_{k,X,\lambda}(x)} y^1 + \cdots + \frac{h^N_{k,X,\lambda}(x)}{\sum_{i=1}^N h^i_{k,X,\lambda}(x)} y^N, \tag{32}$$

where $h^i_{k,X,\lambda}(x) / \sum_{i=1}^N h^i_{k,X,\lambda}(x)$ represents kernel-smoothed relative membership of x to the set $\{x^i\}$. The regression model $A : \mathcal{X} \to \text{aff}(\{y^1, \cdots, y^N\})$ maps a point $x \in \mathcal{X}$ onto the affine hull of $\{y^1, \cdots, y^N\}$ through an affine combination where the coefficients are computed from the functions in RKHS $\mathcal{H}_k(\mathcal{X})$. The model A is referred to as a kernel affine hull machine in this study.

3.4 An Affine Hull Machine

For a finite set $\{y^1, \cdots, y^N\} \subset \mathbb{R}^p$ of N pairwise distinct points, we aim to learn representation of data points in RKHS. For this, we consider a special case of the regression model $A : \mathcal{X} \to \text{aff}(\{y^1, \cdots, y^N\})$ (defined as in (32)) for $\mathcal{X} = \{Py \mid y \in \mathbb{R}^p\}$, where $P \in \mathbb{R}^{n \times p}$ $(n \leq p)$ is an encoding matrix such that product Py is a lower-dimensional encoding for y. In this case, the indicator function $\mathbb{1}_{\{Py^i\}}$ is approximated through a function in RKHS fitted on the ordered pairs $\{\big(Py^j, \mathbb{1}_{\{Py^i\}}(Py^j)\big) \mid j \in \{1, \cdots, N\}\}$. This leads to the development of a kernel affine hull machine defined formally in Definition 1.

Definition 1 (Kernel Affine Hull Machine (KAHM))**.** *Given a finite number of samples: $Y = [\, y^1 \ \cdots \ y^N \,]^T$ with $y^1, \cdots, y^N \in \mathbb{R}^p$ and a subspace dimension $n \leq p$; a kernel affine hull machine $\mathcal{A}_{Y,n} : \mathbb{R}^p \to \text{aff}(\{y^1, \cdots, y^N\})$ maps an arbitrary point $y \in \mathbb{R}^p$ onto the affine hull of $\{y^1, \cdots, y^N\}$ such that*

$$\mathcal{A}_{Y,n}(y) := \frac{h^1_{k_\theta, YP^T, \lambda^*}(Py)}{\sum_{i=1}^N h^i_{k_\theta, YP^T, \lambda^*}(Py)} y^1 + \cdots + \frac{h^N_{k_\theta, YP^T, \lambda^*}(Py)}{\sum_{i=1}^N h^i_{k_\theta, YP^T, \lambda^*}(Py)} y^N. \tag{33}$$

Here,

- *$P \in \mathbb{R}^{n \times p}$ $(n \leq p)$ is an encoding matrix such that product Py is a lower-dimensional encoding for y. For a given subspace dimension n, P is defined by setting the $i-$th row of P as equal to transpose of eigenvector corresponding to $i-$th largest eigenvalue of sample covariance matrix of dataset $\{y^1, \cdots, y^N\}$.*

- $k_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a positive-definite real-valued kernel on* $\mathcal{X}$ *with a corresponding reproducing kernel Hilbert space* $\mathcal{H}_{k_\theta}(\mathcal{X})$ *where*

$$\mathcal{X} = \{Py \mid y \in \mathbb{R}^p\}. \tag{34}$$

The kernel function k_θ *is chosen of Gaussian type:*

$$k_\theta(x^i, x^j) := \exp\left(-\frac{1}{2n}(x^i - x^j)^T\theta^{-1}(x^i - x^j)\right) \tag{35}$$

where θ *is sample covariance matrix of dataset* $\{Py^1, \cdots, Py^N\}$ *defined as*

$$\theta = \frac{1}{N-1}P\left(Y - \mathbf{1}_N\frac{\sum_{i=1}^N (y^i)^T}{N}\right)^T\left(Y - \mathbf{1}_N\frac{\sum_{i=1}^N (y^i)^T}{N}\right)P^T. \tag{36}$$

- *The function* $h^i_{k_\theta, YP^T, \lambda} : \mathcal{X} \to \mathbb{R}$*, such that* $h^i_{k_\theta, YP^T, \lambda} \in \mathcal{H}_{k_\theta}(\mathcal{X})$*, approximates the indicator function* $\mathbb{1}_{\{Py^i\}} : \mathcal{X} \to \{0, 1\}$ *as the solution of following kernel regularized least squares problem:*

$$h^i_{k_\theta, YP^T, \lambda} = \arg\min_{f \in \mathcal{H}_{k_\theta}(\mathcal{X})}\left(\sum_{j=1}^N \left|\mathbb{1}_{\{Py^i\}}(Py^j) - f(Py^j)\right|^2 + \lambda\|f\|^2_{\mathcal{H}_k(\mathcal{X})}\right),\ \lambda \in \mathbb{R}_+. \tag{37}$$

The solution follows as

$$h^i_{k_\theta, YP^T, \lambda}(\cdot) = (I_N)_{i,:}\left(K_{YP^T} + \lambda I_N\right)^{-1}\left[\, k_\theta(\cdot, Py^1)\ \cdots\ k_\theta(\cdot, Py^N)\,\right]^T \tag{38}$$

where $(I_N)_{i,:}$ *denotes the* $i-$*th row of identity matrix of size* N *and* K_{YP^T} *is* $N \times N$ *kernel matrix with its* $(i, j)-$*th element defined as*

$$(K_{YP^T})_{ij} := k_\theta(Py^i, Py^j). \tag{39}$$

The value $h^i_{k_\theta, YP^T, \lambda}(Py)$ *represents the kernel-smoothed membership of point* Py *to the set* $\{Py^i\}$*.*

- *The regularization parameter* $\lambda^* \in \mathbb{R}_+$ *is given as*

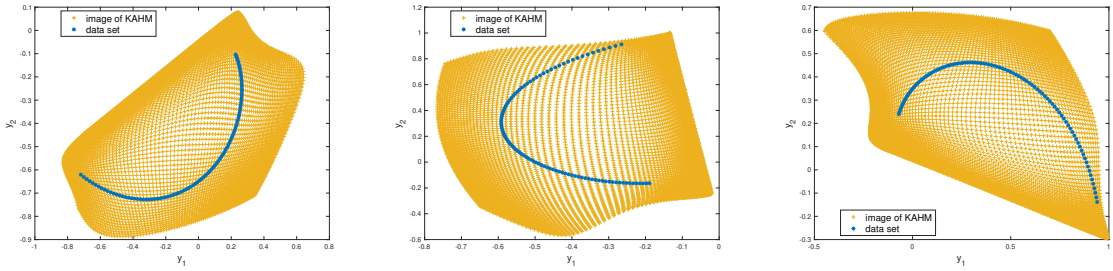
$$\lambda^* = \hat{e} + \frac{2}{pN}\|Y\|_F^2, \tag{40}$$

where $\hat{e}$ *is the unique fixed point of* $\mathcal{R}_{k_\theta, YP^T, Y}$ *such that*

$$\hat{e} = \mathcal{R}_{k_\theta, YP^T, Y}(\hat{e}, \frac{2}{pN}\|Y\|_F^2), \tag{41}$$

with $\mathcal{R}_{k_\theta, YP^T, Y} : \mathbf{R}_+ \times \mathbf{R}_+ \to \mathbf{R}_+$ *defined as*

$$\mathcal{R}_{k_\theta, YP^T, Y}(e, \tau) := \frac{1}{pN}\sum_{j=1}^p \|(Y)_{:,j} - K_{YP^T}\left(K_{YP^T} + (e + \tau)I_N\right)^{-1}(Y)_{:,j}\|^2. \tag{42}$$



(a) first example (b) second example (c) third example

Figure 4: A few examples of two dimensional data sets and KAHM images.

The following iterations

$$e|_{it+1} = \mathcal{R}_{k_\theta, YP^T, Y}(e|_{it}, \frac{2}{pN}\|Y\|_F^2),\ it \in \{0, 1, \cdots\} \tag{43}$$

$$e|_0 \in (0, \frac{1}{pN}\|Y\|_F^2) \tag{44}$$

converge to $\hat{e}$.

- *The image of* $\mathcal{A}_{Y,n}$ *defines a region in the affine hull of* $\{y^1, \cdots, y^N\}$. *That is,*

$$\mathcal{A}_{Y,n}[\mathbb{R}^p] := \{\mathcal{A}_{Y,n}(y) \mid y \in \mathbb{R}^p\} \subset \text{aff}(\{y^1, \cdots, y^N\}). \tag{45}$$

Fig. 4 provides examples of two dimensional datasets and KAHM images.

Remark 2 (Computational complexity)**.** *The computational complexity of the KAHM is asymptotically dominated by the computation of inverse of the* $N \times N$ *dimensional matrix* $(K_{YP^T} + \lambda I_N)$. *Therefore, computational complexity of the KAHM is given as* $\mathcal{O}(N^3)$.

KAHM is a bounded function as stated in Theorem 2 in the following.

Theorem 2. *The KAHM* $\mathcal{A}_{Y,n}$, *associated to* $Y = [\, y^1 \ \cdots \ y^N \,]^T$ *with* $y^1, \cdots, y^N \in \mathbb{R}^p$, *is a bounded function on* $\mathbb{R}^p$ *such that for any* $y \in \mathbb{R}^p$,

$$\|\mathcal{A}_{Y,n}(y)\| < \|Y\|_2 \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} < \|Y\|_2 \left(1 + \frac{pN^2}{2\|Y\|_F^2}\right) \tag{46}$$

where $\lambda^* \in \mathbb{R}_+$ *is defined as in (40) and* K_{YP^T} *is defined as in (39). Thus, the image of* $\mathcal{A}_{Y,n}$ *is bounded such that*

$$\mathcal{A}_{Y,n}[\mathbb{R}^p] \subset \left\{ y \in \mathbb{R}^p \mid \|y\| < \|Y\|_2 \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} \right\}. \tag{47}$$

Proof. The proof is provided in Appendix B. □

A distance function can be associated to KAHM as in Definition 2:

Definition 2 (A Distance Function Induced by KAHM). *Given a KAHM $\mathcal{A}_{Y,n}$, the distance of an arbitrary point $y \in \mathbb{R}^p$ from its image under $\mathcal{A}_{Y,n}$ is given as*

$$\Gamma_{\mathcal{A}_{Y,n}}(y) := \|y - \mathcal{A}_{Y,n}(y)\| . \tag{48}$$

A significant property of the distance function is that its value at a point can not be arbitrary large provided that the point is *sufficiently* close to the samples represented by KAHM. This property is being stated by Theorem 3 in the following.

Theorem 3. *The ratio of the distance of a point $y \in \mathbb{R}^p$ from its image under $\mathcal{A}_{Y,n}$ to the distance of y from $\{y^1, \cdots, y^N\}$ evaluated as $\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2$ remains upper bounded as*

$$\frac{\Gamma_{\mathcal{A}_{Y,n}}(y)}{\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2} < \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} < 1 + \frac{pN^2}{2\|Y\|_F^2} \tag{49}$$

where $\lambda^* \in \mathbb{R}_+$ is defined as in (40), K_{YP^T} is defined as in (39), and $Y = [\, y^1 \ \cdots \ y^N \,]^T$.

Proof. The proof is provided in Appendix C. □

Theorem 3 signifies that if a point y is close to points $\{y^1, \cdots, y^N\}$, then the value $\Gamma_{\mathcal{A}_{Y,n}}(y)$ can not be large. Thus, a large value of the distance function at a point y indicates that y is at far distance from $\{y^1, \cdots, y^N\}$.

4. KAHM for Data Representation Learning

For data representation learning, KAHM based models are introduced (in Section 4.1) and applied to the classification problem (in Section 4.2). To evaluate the risk of membership inference attack through KAHM based classifier, a membership-inference score is defined (in Section 4.3).

4.1 Wide and Conditionally Deep KAHMs

Definition 3 (Conditionally Deep Kernel Affine Hull Machine). *Given a finite number of samples: $Y = [\, y^1 \ \cdots \ y^N \,]^T$ with $y^1, \cdots, y^N \in \mathbb{R}^p$, a subspace dimension $n \leq p$, and number of layers $L \leq n$; a conditionally deep kernel affine hull machine $\mathcal{D}_{Y,n,L} : \mathbb{R}^p \to \text{aff}(\{y^1, \cdots, y^N\})$ maps an arbitrary point $y \in \mathbb{R}^p$ onto the affine hull of $\{y^1, \cdots, y^N\}$ through a nested composition of kernel affine hull machines (as illustrated in Fig. 5) such that*

$$\mathcal{D}_{Y,n,L}(y) = \mathcal{M}_{Y,n,\hat{l}(y)}(y), \tag{50}$$

$$\mathcal{M}_{Y,n,l}(y) = (\mathcal{A}_{Y,n-l+1} \circ \cdots \circ \mathcal{A}_{Y,n-1} \circ \mathcal{A}_{Y,n})\,(y), \tag{51}$$

$$\hat{l}(y) = \arg \min_{l \in \{1,2,\cdots,L\}} \|y - \mathcal{M}_{Y,n,l}(y)\| , \tag{52}$$

where $\mathcal{A}_{Y,.}$ is a KAHM (Definition 1) and $\mathcal{M}_{Y,n,l}(y)$ is the image of y onto the affine hull of $\{y^1, \cdots, y^N\}$ by the $l-th$ layer and the output $\mathcal{D}_{Y,n,L}(y)$ is equal to the image of y onto the affine hull of $\{y^1, \cdots, y^N\}$ by $\hat{l}-th$ layer (which is the layer resulting in minimum Euclidean distance between input vector y and its image onto the affine hull of $\{y^1, \cdots, y^N\}$). The

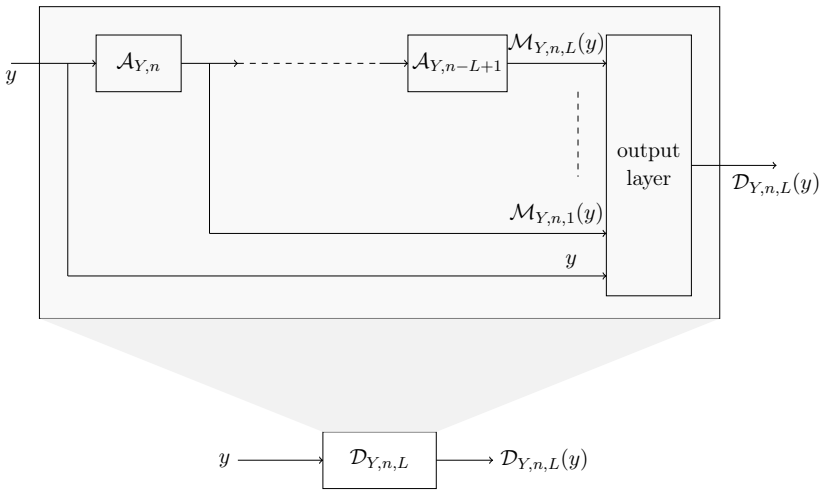


Figure 5: The structure of conditionally deep $L-$layered kernel affine hull machine.

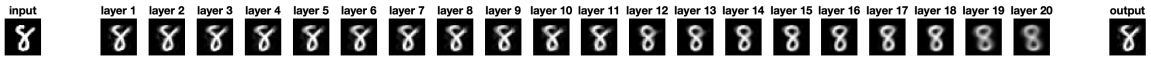


Figure 6: A dataset Y consisting of 1000 randomly chosen samples of digit 8 from MNIST dataset was considered. Corresponding to an input sample y (displayed at extreme left of the figure), the outputs of different layers (i.e. $\mathcal{M}_{Y,n,1}(y), \cdots, \mathcal{M}_{Y,n,20}(y)$) have been displayed. The output of conditionally deep KAHM (i.e. $\mathcal{D}_{Y,n,20}(y)$) has been displayed at extreme right of the figure.

deep KAHM discovers layers of increasingly abstract data representation with lowest-level data features being modeled by first layer and the highest-level by end layer. Fig. 6 illustrates through an example the data representation learning at varying abstraction levels across different layers such that $\mathcal{M}_{Y,n,1}(y)$ is least abstract representation and $\mathcal{M}_{Y,n,L}(y)$ is most abstract representation of the input vector y.

Definition 4 (A Distance Function Induced by Conditionally Deep KAHM)**.** *Given a conditionally deep KAHM $\mathcal{D}_{Y,n,L}$, the distance of an arbitrary point $y \in \mathbb{R}^p$ from its image under $\mathcal{D}_{Y,n,L}$ is given as*

$$\Gamma_{\mathcal{D}_{Y,n,L}}(y) := \|y - \mathcal{D}_{Y,n,L}(y)\|\,. \tag{53}$$

Theorem 4. *The ratio of the distance of a point $y \in \mathbb{R}^p$ from its image under $\mathcal{D}_{Y,n,L}$ to the distance of y from $\{y^1, \cdots, y^N\}$ evaluated as $\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2$ remains upper bounded as*

$$\frac{\Gamma_{\mathcal{D}_{Y,n,L}}(y)}{\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2} \leq \frac{\Gamma_{\mathcal{A}_{Y,n}}(y)}{\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2} < 1 + \frac{pN^2}{2\|Y\|_F^2} \tag{54}$$

where $Y = [\, y^1 \ \cdots \ y^N \,]^T$.

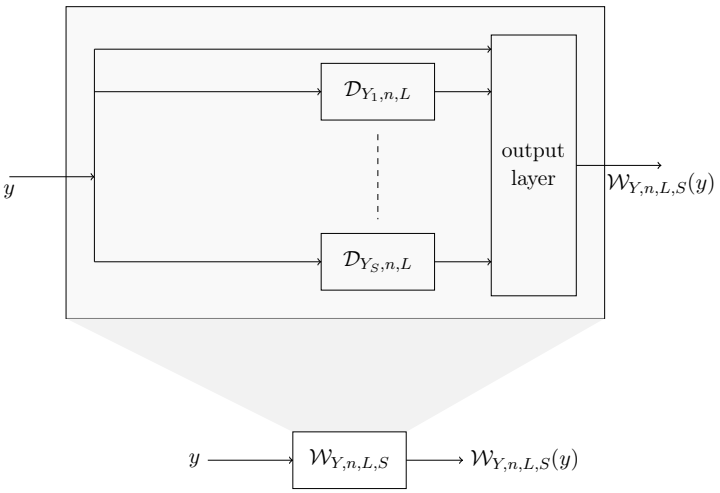


Figure 7: The structure of $S-$branches wide conditionally deep $L-$layered KAHM.

Proof. The proof is provided in Appendix D. □

For big datasets, the total data can be partitioned into subsets and corresponding to each data-subset a separate KAHM can be built to avoid computational challenges associated to big datasets. This motivates to introduce a wide condition deep KAHM in the following:

Definition 5 (Wide Conditionally Deep Kernel Affine Hull Machine)**.** *Given a big but finite number of samples: $Y = [\, y^1 \ \cdots \ y^N \,]^T$ with $y^1, \cdots, y^N \in \mathbb{R}^p$, a subspace dimension $n \leq p$, number of layers $L \leq n$, and number of branches $S \leq N$; a wide conditionally deep kernel affine hull machine $\mathcal{W}_{Y,n,L,S} : \mathbb{R}^p \to \text{aff}(\{y^1, \cdots, y^N\})$ maps an arbitrary point $y \in \mathbb{R}^p$ onto the affine hull of $\{y^1, \cdots, y^N\}$ through a parallel composition of conditionally deep $L-$layered kernel affine hull machines (as illustrated in Fig. 7) such that*

$$\mathcal{W}_{Y,n,L,S}(y) = \mathcal{D}_{Y_{\hat{s}(y)},n,L}(y), \tag{55}$$

$$\hat{s}(y) = \arg \min_{s \in \{1,2,\cdots,S\}} \left\| y - \mathcal{D}_{Y_s,n,L}(y) \right\|, \tag{56}$$

$$Y_s = [\, y^{1,1} \ \cdots \ y^{N_s,s} \,]^T \tag{57}$$

$$\left\{ \{y^{1,1}, \cdots, y^{N_1,1}\}, \cdots, \{y^{1,S}, \cdots, y^{N_S,S}\} \right\} = \text{clustering}(\{y^1, \cdots, y^N\}, S), \tag{58}$$

where $\mathcal{D}_{Y_\cdot,n,L}$ is the conditionally deep KAHM (Definition 3) and clustering$(\{y^1, \cdots, y^N\}, S)$ *represents $k-$means clustering into S subsets, where S can be chosen e.g. as equal to rounding of $N/1000$ towards nearest integer i.e.*

$$S = \lceil N/1000 \rceil. \tag{59}$$

Each of S data clusters leads to a separate conditionally deep KAHM and the output $\mathcal{W}_{Y,n,L,S}(y)$ is equal to the image of y onto the affine hull of $\{y^1, \cdots, y^N\}$ by $\hat{s}-$th conditionally deep KAHM (which is the KAHM resulting in minimum Euclidean distance between input vector y and its image onto the affine hull of $\{y^1, \cdots, y^N\}$).

Definition 6 (A Distance Function Induced by Wide Conditionally Deep KAHM)**.** *Given a wide conditionally deep KAHM $\mathcal{W}_{Y,n,L,S}$, the distance of an arbitrary point $y \in \mathbb{R}^p$ from its image under $\mathcal{W}_{Y,n,L,S}$ is given as*

$$\Gamma_{\mathcal{W}_{Y,n,L,S}}(y) := \|y - \mathcal{W}_{Y,n,L,S}(y)\|. \tag{60}$$

Theorem 5. *The ratio of the distance of a point $y \in \mathbb{R}^p$ from its image under $\mathcal{W}_{Y,n,L,S}$ to the distance of y from $\{y^1, \cdots, y^N\}$ evaluated as $\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_F$ remains upper bounded as*

$$\frac{\Gamma_{\mathcal{W}_{Y,n,L,S}}(y)}{\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_F} < \min_{s \in \{1,2,\cdots,S\}} \left(1 + \frac{pN_s^2}{2\|Y_s\|_F^2}\right) \tag{61}$$

where Y_s is given as in (57).

Proof. The proof is provided in Appendix E. □

4.2 Classification Applications

The KAHM induced distance function, with a property as stated in Theorem 5, can be leveraged to define a classifier. The significance of inequality (61) is that if a data point $y \in \mathbb{R}^p$ is close to samples $\{y^1, \cdots, y^N\}$, then the value $\Gamma_{\mathcal{W}_{Y,n,L,S}}(y)$ remains small. This allows to define a classifier, as in Definition 7, via modeling each class's region through a separate wide conditionally deep KAHM and assigning to a point the label of the class with the minimum distance function value.

Definition 7 (KAHM Based Classifier)**.** *Given a multi-class labelled dataset $\{(Y_i, \mathrm{cl}^i) \mid Y_i = [\, y^{1,i} \ \cdots \ y^{N_i,i} \,]^T,\ y^{\cdot,i} \in \mathbb{R}^p,\ \mathrm{cl}^i \in \{1, 2, \cdots, C\},\ i \in \{1, 2, \cdots, C\}\}$, a KAHM based classifier $\mathcal{C} : \mathbb{R}^p \to \{1, 2, \cdots, C\}$ is defined as*

$$\mathcal{C}(y; \mathcal{W}_{Y_1,n,L,S_1}, \cdots, \mathcal{W}_{Y_C,n,L,S_C}) = \arg \min_{c \in \{1,2,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,n,L,S_c}}(y), \tag{62}$$

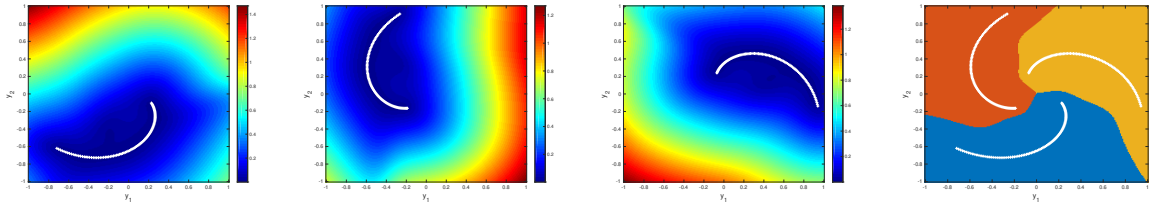
where $\mathcal{W}_{Y_c,n,L,S_c}$ is the wide conditionally deep KAHM (Definition 5) modeling the $c-th$ class labelled data points and $\Gamma_{\mathcal{W}_{Y_c,n,L,S_c}}(\cdot)$ is the distance function (Definition 6) induced by $\mathcal{W}_{Y_c,n,L,S_c}$. The classifier assigns to an arbitrary point $y \in \mathbb{R}^p$ the label of the class which has the minimum distance between y and y's image onto the affine hull of samples of that class. Fig. 8 shows an example of a KAHM based classifier built with a 2-dimensional dataset with 3 classes.

The distance function can be further used to define a class-matching score as in Definition 8.

Definition 8 (Class-Matching Score)**.** *Given the set $\{\mathcal{W}_{Y_c,n,L,S_c}\}_{c=1}^C$ (where $\mathcal{W}_{Y_c,n,L,S_c}$ is the wide conditionally deep KAHM (Definition 5) modeling the $c-th$ class labelled data points), the matching-score of a point $y \in \mathbb{R}^p$ to $c-th$ class is defined as*

$$ms(y; \mathcal{W}_{Y_1,n,L,S_1}, \cdots, \mathcal{W}_{Y_C,n,L,S_C}) = \exp\left(-\frac{|\Gamma_{\mathcal{W}_{Y_c,n,L,S_c}}(y)|^2}{\sum_{c=1}^C |\Gamma_{\mathcal{W}_{Y_c,n,L,S_c}}(y)|^2}\right) \tag{63}$$

where $\Gamma_{\mathcal{W}_{Y_c,n,L,S_c}}(\cdot)$ is the distance function (Definition 6) induced by $\mathcal{W}_{Y_c,n,L,S_c}$.



(a) Y_1 and $\Gamma_{\mathcal{W}_{Y_1,2,5,1}}(\cdot)$ (b) Y_2 and $\Gamma_{\mathcal{W}_{Y_2,2,5,1}}(\cdot)$ (c) Y_3 and $\Gamma_{\mathcal{W}_{Y_3,2,5,1}}(\cdot)$ (d) decision boundaries determined by the classifier (62)

Figure 8: An example of KAHM based classifier built with a 2-dimensional dataset with 3 classes. The data samples have been displayed using '+' marker and the distance function $\Gamma_{\mathcal{W}_{Y,n,L,S}}(\cdot)$ has been displayed as color plot.

4.3 Membership-Inference Score For KAHM Based Classifier

The KAHM based classifier (Definition 7) is built using the training dataset,

$$\mathbf{D}_{trn} = \{y^{j,i} \mid j \in \{1,2,\cdots,N_i\},\ i \in \{1,2,\cdots,C\}\}. \tag{64}$$

As observed in (62), the classifier assigns a label to a vector y based on distance function values: $\{\Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}(y)\}_{c=1}^{C}$. It is obvious that a point either belonging to or lying close to points represented by matrix Y_c (i.e. $\{y^{1,c},\cdots,y^{N_c,c}\}$) will have the value of distance function $\Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}$ smaller than the value corresponding to a point lying away from points $\{y^{1,c},\cdots,y^{N_c,c}\}$. So the distance function $\Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}$ carries an information about the membership of a point to the set of points represented by Y_c. Similarly, the value $\min_{c\in\{1,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}(y)$ carries an information about the membership of y to the training dataset $\mathbf{D}_{trn}$. To evaluate the potential of function $\min_{c\in\{1,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}$ in inferring the membership of a data point to training dataset $\mathbf{D}_{trn}$, a score, referred to as "membership-inference score", is defined. For this we define, for a given small positive number $\mathrm{o} \in \mathbb{R}_{\geq 0}$, sets $\mathbf{T}_\mathrm{o}, \mathbf{T}'_\mathrm{o} \subset \mathbb{R}_{\geq 0}$ as

$$\mathbf{T}_\mathrm{o} = \{y \in \mathbb{R}^p \mid \min_{y'\in\mathbf{D}_{trn}} \|y - y'\| \leq \mathrm{o}\},\ \mathrm{o} \in \mathbb{R}_{\geq 0}, \tag{65}$$

$$\mathbf{T}'_\mathrm{o} = \{y \in \mathbb{R}^p \mid \min_{y'\in\mathbf{D}_{trn}} \|y - y'\| > \mathrm{o}\},\ \mathrm{o} \in \mathbb{R}_{\geq 0}. \tag{66}$$

Further define two non-negative functions, $\mathbf{r}_\mathrm{o} : \mathbf{T}_\mathrm{o} \to \mathbb{R}_{\geq 0}$ and $\mathbf{r}'_\mathrm{o} : \mathbf{T}'_\mathrm{o} \to \mathbb{R}_{\geq 0}$, as

$$\mathbf{r}_\mathrm{o}(y) = \min_{c\in\{1,2,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,n,L,S_c}}(y),\ y \in \mathbf{T}_\mathrm{o}, \tag{67}$$

$$\mathbf{r}'_\mathrm{o}(y') = \min_{c\in\{1,2,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,n,L,S_c}}(y'),\ y' \in \mathbf{T}'_\mathrm{o}. \tag{68}$$

Let $f_{\mathbf{r}_\mathrm{o}}$ and $f_{\mathbf{r}'_\mathrm{o}}$ denote the densities of probability distributions on $\mathbf{r}_\mathrm{o}$ and $\mathbf{r}'_\mathrm{o}$ respectively. It is obvious that $f_{\mathbf{r}_\mathrm{o}}$ characterizes the distribution of values taken by the function $\min_{c\in\{1,2,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}$ over data points lying within the distance of o from any data

point included in the set $\mathbf{D}_{trn}$. Similarly, $f_{\mathbf{r}'_{\mathrm{o}}}$ characterizes the distribution of values taken by the function $\min_{c\in\{1,2,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}$ over data points lying away from the training dataset. Thus, the difference between $f_{\mathbf{r}_{\mathrm{o}}}$ and $f_{\mathbf{r}'_{\mathrm{o}}}$ is a measure of the potential of function $\min_{c\in\{1,\cdots,C\}} \Gamma_{\mathcal{W}_{Y_c,\cdot,\cdot,\cdot}}$ in inferring the membership of a datapoint to training dataset $\mathbf{D}_{trn}$. Hence, the membership-inference score is defined as the $L2$ distance between $f_{\mathbf{r}_{\mathrm{o}}}$ and $f_{\mathbf{r}'_{\mathrm{o}}}$:

$$mis := \int (f_{\mathbf{r}_{\mathrm{o}}}(r) - f_{\mathbf{r}'_{\mathrm{o}}}(r))^2 \,\mathrm{d}r. \tag{69}$$

Taking $\mathrm{o} = 0$, training dataset $\mathbf{D}_{trn}$ can be used to generate samples from $f_{\mathbf{r}_0}$, i.e., $\{\mathbf{r}_0(y) \mid y \in \mathbf{D}_{trn}\}$ is the set of samples generated from $f_{\mathbf{r}_0}$. Similarly, test dataset (which is typically used to evaluate the classifier's performance), $\mathbf{D}_{tst}$, can be used to generate samples from $f'_{\mathbf{r}_0}$, i.e., $\{\mathbf{r}'_0(y') \mid y' \in \mathbf{D}_{tst}\}$ is the set of samples generated from $f'_{\mathbf{r}_0}$. Now, the membership-inference score can be computed by approximating the $L2$ distance between $f_{\mathbf{r}_0}$ and $f_{\mathbf{r}'_0}$ from the samples $\{\mathbf{r}_0(y) \mid y \in \mathbf{D}_{trn}\}$ and $\{\mathbf{r}'_0(y') \mid y' \in \mathbf{D}_{tst}\}$ using a density-difference estimation method (Sugiyama et al., 2013).

5. Privacy-Preserving Learning

Assuming that training dataset is private, KAHM based classification problem is considered under differential privacy framework. For this, an optimal differentially private noise adding mechanism is reviewed (in Section 5.1) and a novel differentially private data fabrication method is developed for classification applications (in Section 5.2). The application of KAHM to differentially private federated learning is considered (in Section 5.3).

5.1 An Optimal $(\epsilon, \delta)-$Differentially Private Noise Adding Mechanism

A given computational algorithm, operating on a data matrix $Y \in \mathbb{R}^{N\times p}$, can be represented by a mapping, $alg : \mathbb{R}^{N\times p} \to Range(alg)$. The privacy of data matrix Y can be preserved via adding a suitable random noise to Y before the application of algorithm alg on the data matrix. This will result in a private version of algorithm alg which is formally defined by Definition 9.

Definition 9 (A Private Algorithm on a Data Matrix)**.** *Given a computational algorithm $alg : \mathbb{R}^{N\times p} \to Range(alg)$, a private version of alg, $alg^+ : \mathbb{R}^{N\times p} \to Range(alg^+)$, is defined as*

$$alg^+(Y) := alg\left(Y^+\right), \tag{70}$$

$$Y^+ = Y + \mathrm{V},\ \mathrm{V} \in \mathbb{R}^{N\times p}, \tag{71}$$

where V *is a random noise matrix with $f_{\mathrm{v}^i_j}(v)$ being the probability density function of its $(i,j)-$th element v^i_j; v^i_j and $\mathrm{v}^{i'}_j$ are independent from each other for $i \neq i'$; and $alg : \mathbb{R}^{N\times p} \to Range(alg)$ is a given mapping representing a computational algorithm. The range of alg^+ is as*

$$Range(alg^+) = \left\{alg\left(Y + \mathrm{V}\right) \mid Y \in \mathbb{R}^{N\times p}, \mathrm{V} \in \mathbb{R}^{N\times p}\right\}. \tag{72}$$

We consider a threat scenario that an adversary seeks to gain an information about the data matrix Y from an analysis of the change in output of algorithm alg as a result of a change in data matrix. In particularly, we seek to attain differential privacy for algorithm alg^+ against the perturbation in an element of Y, say $(i_0, j_0)-$th element, such that magnitude of the perturbation is upper bounded by a scalar d.

Definition 10 ($d-$Adjacency for Data Matrices). *Two matrices $Y, Y' \in \mathbb{R}^{N\times p}$ are $d-$adjacent if for a given $d \in \mathbb{R}_+$, there exist $i_0 \in \{1, 2, \cdots, N\}$ and $j_0 \in \{1, 2, \cdots, p\}$ such that $\forall i \in \{1, 2, \cdots, N\},\ j \in \{1, 2, \cdots, p\}$,*

$$\left|(Y)_{i,j} - (Y')_{i,j}\right| \leq \begin{cases} d, & \text{if } i = i_0, j = j_0 \\ 0, & \text{otherwise} \end{cases}$$

where $(Y)_{i,j}$ and $(Y')_{i,j}$ denote the $(i, j)-$th element of Y and Y' respectively. Thus, Y and Y' differ by only one element and the magnitude of the difference is upper bounded by d.

Definition 11 ($(\epsilon, \delta)-$Differential Privacy for alg^+ (Kumar et al., 2019)). *The algorithm $alg^+(Y)$ is $(\epsilon, \delta)-$differentially private if*

$$Pr\{alg^+(Y) \in \mathcal{O}\} \leq \exp(\epsilon) Pr\{alg^+\left(Y'\right)) \in \mathcal{O}\} + \delta \tag{73}$$

for any measurable set $\mathcal{O} \subseteq Range(alg^+)$ and for $d-$adjacent matrices pair (Y, Y').

Definition 11 implies that changing the value of an element in the matrix Y by an amount upper bounded by d can change the distribution of output of the algorithm alg^+ only by a factor of $\exp(\epsilon)$ with probability at least $1 - \delta$. Thus, the lower value of ϵ and δ lead to a higher amount of privacy.

Result 1 (An Optimal $(\epsilon, \delta)-$Differentially Private Noise (Kumar et al., 2019)). *The probability density function of noise, that minimizes the expected noise magnitude together with satisfying the sufficient conditions for $(\epsilon, \delta)-$differential privacy for alg^+, is given as*

$$f^*_{\mathrm{v}^i_j}(v; \epsilon, \delta, d) = \begin{cases} \delta\, Dirac\delta(v), & v = 0 \\ (1-\delta)\frac{\epsilon}{2d}\exp(-\frac{\epsilon}{d}|v|), & v \in \mathbb{R}\setminus\{0\} \end{cases} \tag{74}$$

where $Dirac\delta(v)$ is Dirac delta function satisfying $\int_{-\infty}^{\infty} Dirac\delta(v)\ \mathrm{d}v = 1$.

Remark 3 (Generating Random Samples from $f^*_{\mathrm{v}^i_j}$). *The method of inverse transform sampling can be used to generate random samples from cumulative distribution function. The cumulative distribution function of $f^*_{\mathrm{v}^i_j}$ is given as*

$$F_{\mathrm{v}^i_j}(v; \epsilon, \delta, d) = \begin{cases} \frac{1-\delta}{2}\exp(\frac{\epsilon}{d}v), & v < 0 \\ \frac{1+\delta}{2}, & v = 0 \\ 1 - \frac{1-\delta}{2}\exp(-\frac{\epsilon}{d}v), & v > 0 \end{cases} \tag{75}$$

The inverse cumulative distribution function is given as

$$F^{-1}_{\mathrm{v}^i_j}(t^i_j; \epsilon, \delta, d) = \begin{cases} \frac{d}{\epsilon}\log(\frac{2t^i_j}{1-\delta}), & t^i_j < \frac{1-\delta}{2} \\ 0, & t^i_j \in [\frac{1-\delta}{2}, \frac{1+\delta}{2}] \\ -\frac{d}{\epsilon}\log(\frac{2(1-t^i_j)}{1-\delta}), & t^i_j > \frac{1+\delta}{2} \end{cases},\ t^i_j \in (0, 1). \tag{76}$$

Thus, via generating random samples from the uniform distribution on $(0, 1)$ and using (76), the noise additive mechanism can be implemented.

Algorithm 1 Differentially private approximation of a data matrix (Kumar, 2023)

Require: Data matrix $Y \in \mathbb{R}^{N\times p}$; differential privacy parameters: $d \in \mathbb{R}_+$, $\epsilon \in \mathbb{R}_+$, $\delta \in (0,1)$.

1: Compute $\forall\ i \in \{1,2,\cdots,N\},\ j \in \{1,2,\cdots,p\}$,

$$(Y_\epsilon^+)_{i,j} = (Y)_{i,j} + F^{-1}_{\mathrm{v}_j^i}(t_j^i;\epsilon,\delta,d),\ t_j^i \in (0,1), \tag{77}$$

where t_j^i is chosen from the uniform distribution on $(0,1)$ and $F^{-1}_{\mathrm{v}_j^i}$ is given by (76).

2: **return** Y_ϵ^+ (where the subscript ϵ indicates the given privacy-loss bound ϵ).

For a given value of (ϵ,δ,d), Algorithm 1 is stated for the differentially private approximation of a data matrix.

5.2 Differentially Private Data Fabrication and Classification

A computational algorithm can be made to ensure differential privacy (i.e. inequality (73)) via applying the algorithm on the data matrix returned by Algorithm 1. Hence, a KAHM based differentially private classifier can be built as in Definition 12.

Definition 12 (A KAHM Based Differentially Private Classifier)**.** *Given a multi-class labelled differentially private dataset* $\{(Y^+_{\epsilon,i}, \mathrm{cl}^i) \mid Y^+_{\epsilon,i} \in \mathbb{R}^{N_i\times p},\ \mathrm{cl}^i \in \{1,2,\cdots,C\}\}$*, a KAHM based differentially private classifier* $\mathcal{C}:\mathbb{R}^p \to \{1,2,\cdots,C\}$ *is defined as*

$$\mathcal{C}(y;\mathcal{W}_{Y^+_{\epsilon,1},n,L,S_1},\cdots,\mathcal{W}_{Y^+_{\epsilon,C},n,L,S_C}) = \arg\min_{c\in\{1,2,\cdots,C\}} \Gamma_{\mathcal{W}_{Y^+_{\epsilon,c},n,L,S_c}}(y), \tag{78}$$

where $\mathcal{W}_{Y^+_{\epsilon,c},\cdot,\cdot,\cdot}$ *is the wide conditionally deep KAHM (Definition 5) modeling the* $c-th$ *class labelled data points and* $\Gamma_{\mathcal{W}_{Y^+_{\epsilon,c},\cdot,\cdot,\cdot}}(\cdot)$ *is the distance function (Definition 6) induced by* $\mathcal{W}_{Y^+_{\epsilon,c},\cdot,\cdot,\cdot}$*, and* $Y^+_{\epsilon,c}$ *is differentially private data matrix obtained by Algorithm 1. The classifier assigns to an arbitrary point* $y \in \mathbb{R}^p$ *the label of the class which has the minimum distance between* y *and* y*'s image onto the affine hull of differentially private samples of that class.*

Since a differentially private algorithm operates on noise added data, the algorithm's performance is adversely affected. An obvious effect of adding noise to data matrix is an increase in the modeling error of data samples by a KAHM. Typically, we have

$$\sum_{i=1}^{N} \|y_\epsilon^{+i} - \mathcal{A}_{Y^+_\epsilon,n}\left(y_\epsilon^{+i}\right)\| > \sum_{i=1}^{N} \|y^i - \mathcal{A}_{Y,n}\left(y^i\right)\|, \tag{79}$$

where $y_\epsilon^{+i} = ((Y_\epsilon^+)_{i,:})^T$. Thus, an approach to alleviate the effect of added noise on the performance of a KAHM based algorithm is of processing the noise added data matrix through a data smoother such that the smoothed data matrix leads to a KAHM with modeling error not larger than the modeling error on original data samples. One such smoother is defined as in Definition 13.

Definition 13 (A Smoother for Differentially Private Data)**.** *Given a differentially private matrix $Y_\epsilon^+ \in \mathbb{R}^{N\times p}$, a subspace dimension $n \leq p$, and a positive integer $M \in \mathbb{Z}_+$; Y_ϵ^+ is transformed into $\hat{Y}_{M-1} \in \mathbb{R}^{N\times p}$ through following recursions run from $m = 0$ to $m = M-1$:*

$$\hat{y}^{i,0} = ((Y_\epsilon^+)_{i,:})^T,\ \forall i \in \{1, 2, \cdots, N\}, \tag{80}$$

$$\hat{y}^{i,m+1} = \left(\sum_{j=1}^{N} h^j_{k_{\theta_m}, \hat{Y}_m P_m^T, \lambda_m^*}(P_m \hat{y}^{i,m})\right) \times \mathcal{A}_{\hat{Y}_m, n}\left(\hat{y}^{i,m}\right), \tag{81}$$

$$\hat{Y}_m = [\, \hat{y}^{1,m}\ \cdots\ \hat{y}^{N,m}\,]^T, \tag{82}$$

where

- *P_m is defined by setting the $i-$th row of P_m as equal to transpose of eigenvector corresponding to $i-$th largest eigenvalue of sample covariance matrix of the dataset $\{\hat{y}^{1,m}, \cdots, \hat{y}^{N,m}\}$.*
- *θ_m is sample covariance matrix of dataset $\{P_m\hat{y}^{1,m}, \cdots, P_m\hat{y}^{N,m}\}$, i.e.,*

$$\theta_m = \frac{1}{N-1}_m P_m \left(\hat{Y}_m - \mathbf{1}_N \frac{\sum_{i=1}^N (\hat{y}^{i,m})^T}{N}\right)^T \left(\hat{Y}_m - \mathbf{1}_N \frac{\sum_{i=1}^N (\hat{y}^{i,m})^T}{N}\right) P_m^T. \tag{83}$$

- *$\lambda_m^* \in \mathbb{R}_+$ is given as*

$$\lambda_m^* = \hat{e}_m + \frac{2}{pN}\|\hat{Y}_m\|_F^2, \tag{84}$$

  where $\hat{e}_m$ is the unique fixed point of the function $\mathcal{R}_{k_{\theta_m}, \hat{Y}_m P_m^T, \hat{Y}_m}$ (which is defined as in (42)).

- *$k_{\theta_m}(\cdot,\cdot)$ and $h^i_{k_{\theta_m}, \hat{Y}_m P_m^T, \lambda_m^*}(\cdot)$ are defined as in (35) and (38) respectively.*

The transformation of Y_ϵ^+ into $\hat{Y}_{M-1}$ is represented as

$$\hat{Y}_{M-1} = \mathrm{T}_{n,M}(Y_\epsilon^+). \tag{85}$$

The transformation of Y_ϵ^+ into $\hat{Y}_{M-1}$ has been defined in a particular way to ensure a property related to the error in KAHM modeling of data samples. This property is sated in Theorem 6.

Theorem 6. *The error in KAHM modeling of smoothed data matrix $\hat{Y}_{M-1} = \mathrm{T}_{n,M}(Y_\epsilon^+)$ converges asymptotically with an increasing value of M to zero, i.e.,*

$$\lim_{M\to\infty} \sum_{i=1}^{N} \|\hat{y}^{i,M-1} - \mathcal{A}_{\hat{Y}_{M-1}, n}(\hat{y}^{i,M-1})\| = 0. \tag{86}$$

where $\hat{Y}_{M-1} = [\, \hat{y}^{1,M-1}\ \cdots\ \hat{y}^{N,M-1}\,]^T$ is computed using recursions (80-82) from $m = 0$ to $m = M-1$.

Proof. The proof is provided in Appendix F. □

It follows from Theorem 6 that the KAHM modeling error of smoothed data samples can be reduced to an arbitrary low value by choosing a sufficiently large value of M. Thus, it is possible to find the smallest number, say $\tilde{M} \in \mathbb{Z}_+$, ensuring that

$$\sum_{i=1}^{N} \|\hat{y}^{i,\tilde{M}-1} - \mathcal{A}_{\hat{Y}_{\tilde{M}-1},n}(\hat{y}^{i,\tilde{M}-1})\| \leq r, \tag{87}$$

where r is the error in KAHM modeling of original data matrix Y defined as

$$r = \sum_{i=1}^{N} \|y^i - \mathcal{A}_{Y,n}(y^i)\|. \tag{88}$$

That is, error in KAHM modeling of smoothed data matrix $\hat{Y}_{\tilde{M}-1}$ is lower than the error in KAHM modeling of original data matrix Y, which suggests that applying a KAHM based computational algorithm on $\hat{Y}_{\tilde{M}-1}$ (instead of Y_ϵ^+) may alleviate the effect of added noise on the performance of a KAHM based computational algorithm. This motivates to use the KAHM associated to $\hat{Y}_{\tilde{M}-1}$, i.e. $\mathcal{A}_{\hat{Y}_{\tilde{M}-1},n}$, for fabricating data samples meant for building KAHM based models.

Definition 14 (Differentially Private Fabricated Data)**.** *Given a differentially private matrix $Y_\epsilon^+ \in \mathbb{R}^{N\times p}$ ensuring the privacy-loss bound $\epsilon \in \mathbb{R}_+$, a subspace dimension $n \leq p$, and error in KAHM modeling of original data matrix Y evaluated as $r = \sum_{i=1}^{N} \|y^i - \mathcal{A}_{Y,n}(y^i)\|$; a differentially private fabricated data matrix $\tilde{Y} \in \mathbb{R}^{N\times p}$ is defined as*

$$\tilde{Y} = \left[\, \mathcal{A}_{\hat{Y}_{\tilde{M}-1},n}(\hat{y}^{1,\tilde{M}-1}) \;\cdots\; \mathcal{A}_{\hat{Y}_{\tilde{M}-1},n}(\hat{y}^{N,\tilde{M}-1}) \,\right]^T, \tag{89}$$

$$\hat{y}^{i,\tilde{M}-1} = ((\hat{Y}_{\tilde{M}-1})_{i,:})^T, \tag{90}$$

$$\hat{Y}_{\tilde{M}-1} = \mathrm{T}_{n,\tilde{M}}(Y_\epsilon^+), \tag{91}$$

where smoother $\mathrm{T}_{n,\tilde{M}}$ (Definition 13) computes $\hat{Y}_{\tilde{M}-1}$ through recursions (80-82) from $m = 0$ to $m = \tilde{M} - 1$, and $\tilde{M}$ is defined as

$$\tilde{M} = \min \left\{ m \in \mathbb{Z}_+ \;\middle|\; \sum_{i=1}^{N} \|\hat{y}^{i,m-1} - \mathcal{A}_{\hat{Y}_{m-1},n}(\hat{y}^{i,m-1})\| \leq r \right\}. \tag{92}$$

The computing of $\tilde{Y}$ is represented as

$$\tilde{Y} = \mathcal{F}_n(Y_\epsilon^+; r). \tag{93}$$

The fabricated data matrix $\tilde{Y}$ is computed from Y_ϵ^+ (which is a differentially private approximation of Y) and not from original data matrix Y, and thus $\tilde{Y}$ remains differentially private.

Fig. 9 displays a few examples of fabricated data samples corresponding to different choices for privacy-loss bound ϵ and subspace dimension n. As expected and also observed in Fig. 9, more and more features of original data samples get masked in the fabricated data with a decrease in ϵ and/or n.

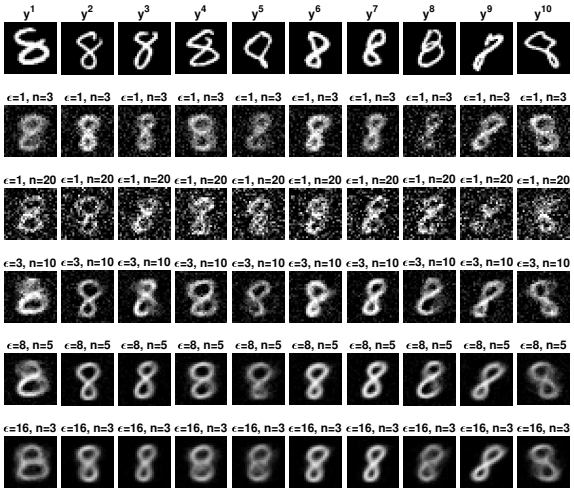


Figure 9: A dataset Y consisting of 1000 randomly chosen samples of digit 8 from MNIST dataset was considered. For 10 randomly selected samples from Y (displayed at top row of the figure), the corresponding samples from differentially private fabricated data $\tilde{Y} = \mathcal{F}_n(Y_\epsilon^+; \sum_{i=1}^{1000} \|y^i - \mathcal{A}_{Y,n}(y^i)\|)$ have been displayed for different values of privacy-loss bound ϵ and subspace dimension n.

Remark 4 (Big Data Fabrication)**.** *For the big datasets with large N, the data can be divided into subsets via e.g. k-means clustering and fabricated data matrix is computed from each subset independently. That is, $\tilde{Y}$ is fabricated as follows:*

$$\tilde{Y} = \Big[\big(\mathcal{F}_n(Y_1^+; r_1)\big)^T \ \cdots \ \big(\mathcal{F}_n(Y_S^+; r_S)\big)^T \Big]^T, \tag{94}$$

$$Y_s^+ \leftarrow \mathit{Algorithm}\ 1(Y_s, d, \epsilon, \delta), \tag{95}$$

$$r_s = \sum_{i=1}^{N_s} \|y^{i,s} - \mathcal{A}_{Y_s,n}(y^{i,s})\|, \tag{96}$$

$$Y_s = [\, y^{1,1} \ \cdots \ y^{N_s,s} \,]^T, \tag{97}$$

$$\{y^{1,s}, \cdots, y^{N_s,s}\}_{s=1}^{S} = \text{clustering}(\{y^1, \cdots, y^N\}, S), \tag{98}$$

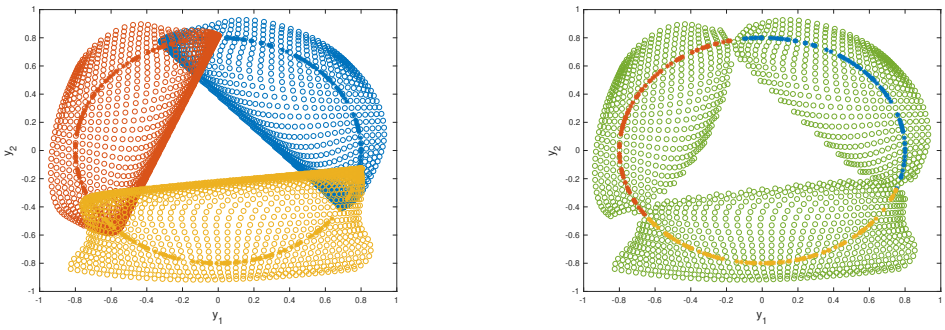
$$S = \lceil N/1000 \rceil, \tag{99}$$

where clustering$(\{y^1, \cdots, y^N\}, S)$ *represents $k-$means clustering into S subsets*

As the fabricated data remain differentially private, a KAHM based classifier can be built using fabricated data to ensure differential privacy in the sense of inequality (73).

Definition 15 (A Differentially Private Classifier Based on Fabricated Data)**.** *Given a multi-class labelled differentially private fabricated dataset $\{(\tilde{Y}_i, \mathrm{cl}^i) \ \mid \ \tilde{Y}_i \in \mathbb{R}^{N_i \times p}, \ \mathrm{cl}^i \in \{1, 2, \cdots, C\}\}$, a classifier $\mathcal{C} : \mathbb{R}^p \to \{1, 2, \cdots, C\}$ is defined as*

$$\mathcal{C}(y; \mathcal{W}_{\tilde{Y}_1,n,L,S_1}, \cdots, \mathcal{W}_{\tilde{Y}_C,n,L,S_C}) = \arg \min_{c \in \{1,2,\cdots,C\}} \Gamma_{\mathcal{W}_{\tilde{Y}_c,n,L,S_c}}(y), \tag{100}$$



(a) images of three different KAHMs built independently using three different datasets

(b) image of the global KAHM combining together independently built KAHMs

Figure 10: An example of combining together local KAHMs to build a global KAHM.

where $\mathcal{W}_{\tilde{Y}_c,\cdot,\cdot,\cdot}$ *is the wide conditionally deep KAHM (Definition 5) modeling the* $c-$*th class labelled data points and* $\Gamma_{\mathcal{W}_{\tilde{Y}_c,\cdot,\cdot,\cdot}}(\cdot)$ *is the distance function (Definition 6) induced by* $\mathcal{W}_{\tilde{Y}_c,\cdot,\cdot,\cdot}$*, and* $\tilde{Y}_c$ *is differentially private fabricated data matrix (Definition 14). The classifier assigns to an arbitrary point* $y \in \mathbb{R}^p$ *the label of the class which has the minimum distance between* y *and* y*'s image onto the affine hull of differentially private fabricated samples of that class.*

5.3 Application to Differentially Private Federated Learning

The multi-class classification problem is considered under the scenario of data being distributed amongst different parties. For the case of data being privately owned by local parties, our federated learning approach is of combining together the local privacy-preserving KAHM based classifiers using the distance functions induced by local KAHMs. For this, a combination of different KAHMs is considered using the distance measure. Given Q different wide conditionally deep KAHMs $\mathcal{W}_{Y^1,n,L,S^1},\cdots,\mathcal{W}_{Y^Q,n,L,S^Q}$ built independently using datasets $Y^1,\cdots,Y^Q$ respectively, a possible way to combine together the KAHMs is as follows:

$$\mathcal{GW}(y;\{\mathcal{W}_{Y^q,n,L,S^q}\}_{q=1}^{Q}) = \mathcal{W}_{Y^{\hat{q}(y)},n,L,S^{\hat{q}(y)}}(y), \tag{101}$$

$$\hat{q}(y) = \arg\min_{q\in\{1,2,\cdots,Q\}} \Gamma_{\mathcal{W}_{Y^q,n,L,S^q}}(y), \tag{102}$$

where $\mathcal{GW}$ is the global KAHM (that combines together the individual KAHMs) and $\Gamma_{\mathcal{W}_{Y^q,n,L,S^q}}$ is the distance function (Definition 6) induced by $\mathcal{W}_{Y^q,n,L,S^q}$. For an input $y \in \mathbb{R}^p$, the global KAHM output is equal to the output of $\hat{q}-$th KAHM (which is the KAHM resulting in minimum Euclidean distance between input vector y and its image onto the affine hull of data samples). A 2-dimensional data example where three different KAHMs are combined to build a global KAHM is provided in Fig. 10. Fig. 10 shows the images of individual KAHMs (in Fig. 10(a)) and the image of global KAHM (in Fig. 10(b)).

The local KAHMs modeling a specific class can be combined together to build a global KAHM (that models the region (in data space) of that class) and a global classifier can be built from all class-specific global KAHMs. Mathematically, the global classifier, $\mathcal{GC}:\mathbb{R}^p \rightarrow$

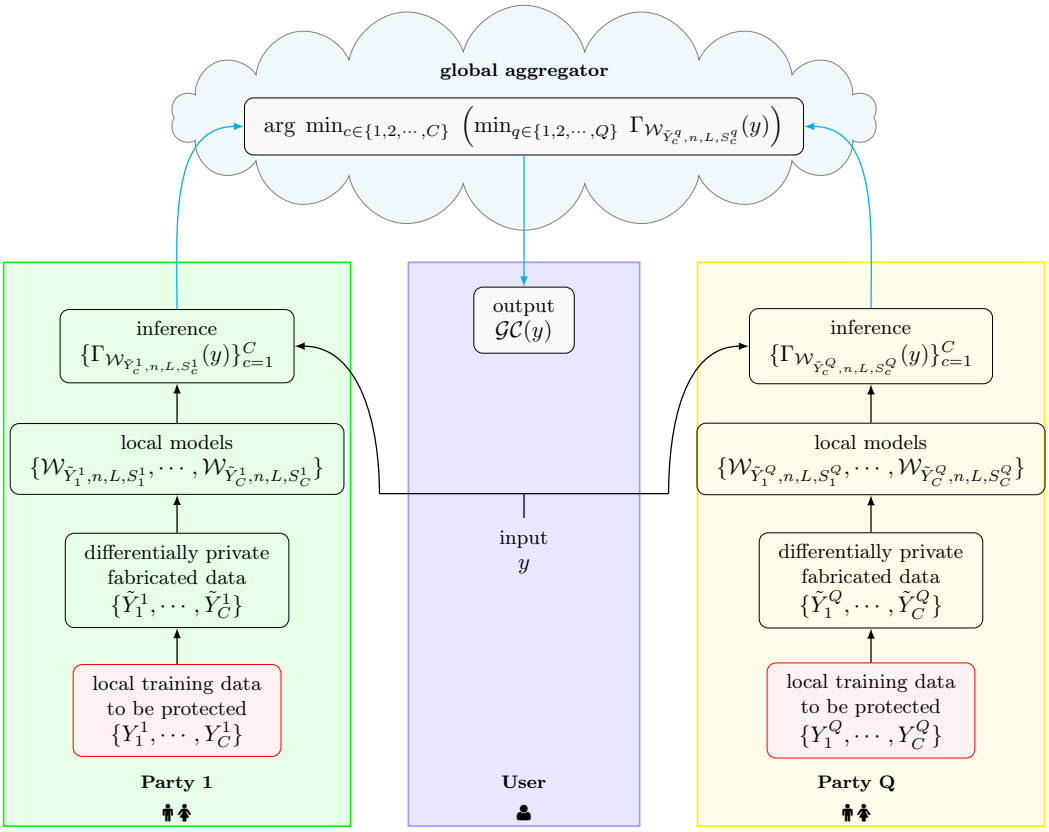


Figure 11: The structural representation of the KAHM based federated learning scheme. The limitation of passing users' inputs to the parties can be addressed as suggested in Remark 6.

$\{1, 2, \cdots, C\}$, is defined as

$$\mathcal{GC}(y) = \arg \min_{c \in \{1,2,\cdots,C\}} \|y - \mathcal{GW}(y; \{\mathcal{W}_{\tilde{Y}_c^q, n, L, S_c^q}\}_{q=1}^{Q})\|, \tag{103}$$

where $\tilde{Y}_c^q$ represents the $c-$th class labelled differentially private data samples fabricated locally by the $q-$th party and $\mathcal{GW}$ is the global KAHM (101). The global classifier (103) assigns to an arbitrary point $y \in \mathbb{R}^p$ the label of the class which has the minimum distance between y and y's image onto the affine hull of differentially private fabricated samples of that class. (103) can be alternatively expressed as

$$\mathcal{GC}(y) = \arg \min_{c \in \{1,2,\cdots,C\}} \left(\min_{q \in \{1,2,\cdots,Q\}} \Gamma_{\mathcal{W}_{\tilde{Y}_c^q, n, L, S_c^q}}(y) \right). \tag{104}$$

An important feature of the global classifier evaluation using (104) is that the evaluation doesn't require individual KAHMs (that are owned by different parties) but only the distance measures. This allows to design a KAHM based differentially private federated learning scheme as illustrated in Fig. 11.

Remark 5 (Local Training Data with Missing Classes)**.** *If the $q-$th party has zero $c-$th labelled data samples, the global classifier (104) is evaluated taking $\Gamma_{\mathcal{W}_{\tilde{Y}_c^q, n, L, S_c^q}}(y) = \infty$.*

Remark 6 (Addressing the Limitation of Passing Users' Inputs to the Clients)**.** *A limitation of the federated learning scheme, as sketched in Fig. 11, is that a user's input query is passed to all of the parties, causing an increased communication cost and concerns regarding the privacy of user's input. This limitation can be easily addressed by transferring all of the local models $\{\{\mathcal{W}_{\tilde{Y}_c^q, n, L, S_c^q}\}_{c=1}^{C}\}_{q=1}^{Q}$ to the cloud.*

6. Experiments

The aim of the experiments is to 1) investigate the performance of KAHM based classifier (in Section 6.1); 2) evaluate the proposed privacy-preserving learning method in-terms of both accuracy and risk of membership inference attack (in Section 6.2); 3) investigate the performance of the proposed differentially private federated learning scheme (in Section 6.3); and 4) study the computational time of KAHM in relation to increasing data dimension, subspace dimension, and data sample size (in Section 6.4).

6.1 KAHM Based Classification of High-Dimensional Feature Vectors

The "Freiburg Groceries Dataset" (Jund et al., 2016) is considered to evaluate the performance of KAHM based classifier (Definition 7). This dataset has around 5000 labeled images of grocery products commonly sold in Germany. The images have been divided into 25 different categories of grocery products. Following the previous studies (Kumar & Freudenthaler, 2020; Kumar et al., 2021a) on this dataset, image features were extracted from "AlexNet" and "VGG-16" networks (which are pre-trained Convolutional Neural Networks). The activations of the fully connected layer "fc6" in AlexNet constitute a $4096-$dimensional feature vector. Also, the activations of the fully connected layer "fc6" in VGG-16 constitute another $4096-$dimensional feature vector. The features extracted by both networks were joined

Table 3: Experiments on 5 different train-test splits of Freiburg groceries data: I

methods	**accuracy (in %) on test images**					
	1	2	3	4	5	mean
KAHM Classifier (Definition 7)	**89.29**	**87.16**	**87.00**	**86.73**	**87.09**	**87.46**
membership-mappings (Kumar et al., 2021a)	87.82	87.06	85.88	85.63	86.19	86.52
nonparametric fuzzy image mapping (Kumar & Freudenthaler, 2020)	88.21	86.64	85.36	85.13	85.79	86.23
Gaussian fuzzy-mapping (Kumar et al., 2021)	83.50	81.52	79.73	79.60	80.48	80.97
SVM (Kumar et al., 2021a)	77.90	79.54	77.17	76.98	76.98	77.71
1-NN (Kumar et al., 2021a)	78.00	77.97	77.38	76.58	76.28	77.24
Back-propagation training of a deep network (Kumar et al., 2021a)	75.25	77.24	72.67	73.37	71.57	74.02
2-NN (Kumar et al., 2021a)	73.48	73.38	70.11	70.05	70.57	71.52
4-NN (Kumar et al., 2021a)	72.50	73.39	68.89	71.16	70.87	71.36
Random Forest (Kumar et al., 2021a)	63.17	62.63	59.47	59.50	59.76	60.90
Naive Bayes (Kumar et al., 2021a)	56.78	56.78	53.74	55.08	56.26	55.73
Ensemble Learning (Kumar et al., 2021a)	38.31	39.35	38.89	37.69	38.34	38.51
Decision Tree (Kumar et al., 2021a)	31.34	30.59	32.14	31.06	30.73	31.17

together to form a 8192−dimensional vector. The feature vectors were scaled along each dimension to take values between -1 and 1.

The authors of (Jund et al., 2016) provide five different train-test splits of images to evaluate the classification performance. For each of the five train-test data splits, training feature vectors of each class are modeled through a separate wide conditionally deep KAHM taking subspace dimension $n = 20$, number of layers $L = 5$, and number of branches S as given in (59). The performance of the proposed KAHM based classifier is compared in Table 3 with previous studies on this dataset. A related application is of detecting the presence of an individual grocery category in an image based on the value of KAHM based

Table 4: Experiments on 5 different train-test splits of Freiburg groceries data: II

methods	**area under ROC curve (averaged per class)**				
	1	2	3	4	5
KAHM (Definition 8)	**0.9901**	**0.9925**	**0.9970**	**0.9969**	**0.9945**
nonparametric fuzzy image mapping (Kumar & Freudenthaler, 2020)	0.9818	0.9775	0.9754	0.9767	0.9761
deep fuzzy nonparametric model (Zhang et al., 2022)	0.9612	0.9574	0.9601	0.9582	0.9531
SVM (Kumar & Freudenthaler, 2020)	0.9806	0.9766	0.9711	0.9777	0.9760
Random Forest (Kumar & Freudenthaler, 2020)	0.9489	0.9510	0.9372	0.9437	0.9466
4-NN (Kumar & Freudenthaler, 2020)	0.9425	0.9336	0.9325	0.9378	0.9280
2-NN (Kumar & Freudenthaler, 2020)	0.9219	0.9125	0.9118	0.9117	0.9048
Naive Bayes (Kumar & Freudenthaler, 2020)	0.8999	0.9100	0.8866	0.9013	0.8908
1-NN (Kumar & Freudenthaler, 2020)	0.8881	0.8802	0.8803	0.8837	0.8752
Ensemble Learning (Kumar & Freudenthaler, 2020)	0.8856	0.8896	0.8813	0.8818	0.8776
Decision Tree (Kumar & Freudenthaler, 2020)	0.6591	0.6473	0.6528	0.6539	0.6443

class-matching score (i.e. Definition 8). To study the application potential of proposed class-matching score, the receiver operating characteristic (ROC) curves are plotted for test images taking a particular image category as positive class. Table 4 reports the performances of different methods evaluated in-term of area under ROC curve. The best performance of the KAHM based classifier on each of the five train-test data splits is observed in Table 3 and Table 4. The proposed KAHM based classifier is more competitive than the previously studied methods on this dataset.

6.2 KAHM Based Differentially Private Classification with Fabricated Data

The proposed KAHM based approach to privacy-preserving classification is studied on different datasets where the performance of the classifier is evaluated in-terms of both accuracy on test data and the value of membership-inference score. The membership-inference score, *mis* (69), is computed using a density-difference estimation method (Sugiyama et al., 2013).

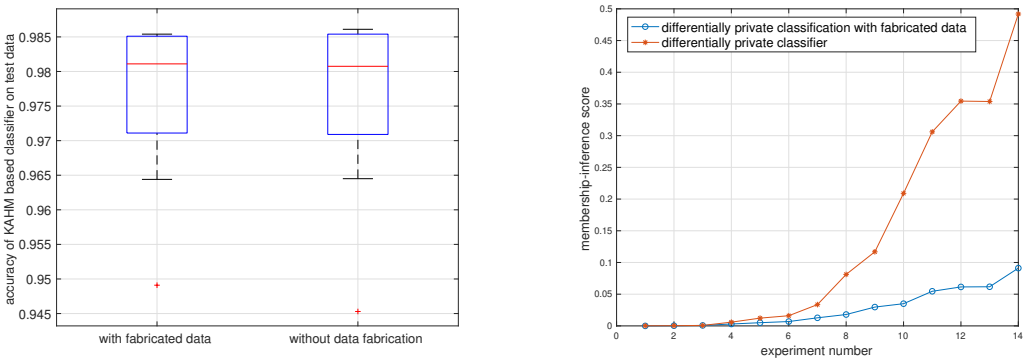
6.2.1 MNIST Dataset

A handwritten digits recognition problem is considered with the widely used MNIST dataset. The dataset contains 28×28 sized images divided into training set of 60000 images and testing set of 10000 images. The images' pixel values are divided by 255 to normalize the values in the range from 0 to 1. The 28×28 normalized values of each image are flattened to an equivalent $784-$dimensional data point.

Table 5: Results of privacy-preserving learning experiments on MNIST dataset

(ϵ, n)	accuracy by classifier (Def. 15)	accuracy by classifier (Def. 12)	*mis* by classifier (Def. 15)	*mis* by classifier (Def. 12)
$(1, 20)$	0.9491	0.9453	0.00000	0.00017
$(1.5, 20)$	0.9644	0.9645	0.00025	0.00035
$(2, 20)$	0.9711	0.9709	0.00074	0.00076
$(3, 20)$	0.9779	0.9776	0.00299	0.00589
$(4, 20)$	0.9802	0.9796	0.00687	0.01605
$(5, 20)$	0.9820	0.9805	0.01275	0.03361
$(8, 20)$	0.9833	0.9845	0.02968	0.11685
$(16, 20)$	0.9854	0.9858	0.05459	0.30601
$(32, 20)$	0.9851	0.9858	0.06171	0.35388
$(32, 5)$	0.9680	0.9676	0.00503	0.01225
$(32, 10)$	0.9799	0.9810	0.01794	0.08117
$(32, 15)$	0.9851	0.9861	0.03501	0.20903
$(32, 20)$	0.9851	0.9854	0.06145	0.35451
$(32, 25)$	0.9845	0.9854	0.09111	0.49186
	0.9772 (mean)	**0.9771** (mean)	**0.02715** (mean)	**0.14160** (mean)

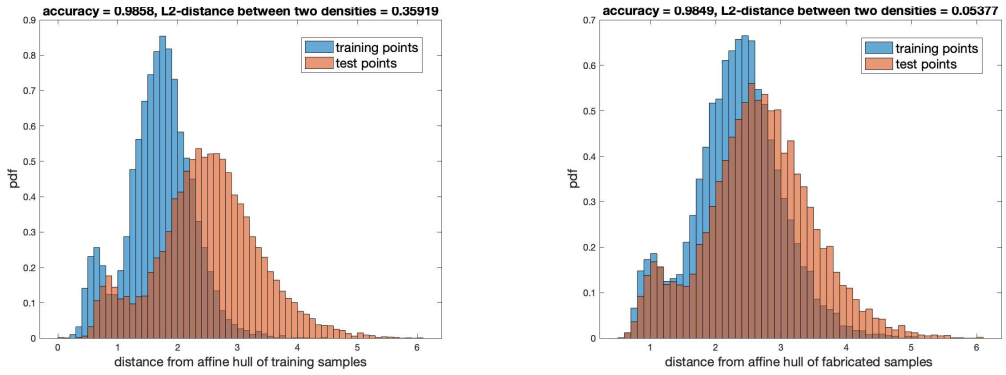
The performances of both differentially private classifier (Definition 12) and differentially private classifier based on fabricated data (Definition 15) are evaluated for different values of privacy-loss bound ϵ and subspace dimension n while keeping the number of layers $L = 5$ and number of branches S as given in (59). Table 5 reports the obtained results. For a visualization of the results, Fig. 12 compares the accuracy and *mis* values obtained by the



(a) comparison of accuracies

(b) comparison of *mis* values

Figure 12: The effect of using fabricated data in classification for MNIST.



(a) histograms of distances of training and test data points from the affine hull of training samples

(b) histograms of distances of training and test data points from the affine hull of fabricated samples

Figure 13: Illustration of the reduction in *mis* value through data fabrication for MNIST.

two methods. It is observed in Fig. 12 that while both classifiers (with and without using fabricated data) achieve nearly the same level of accuracy (as observed in Fig. 12(a)), the membership-inference score is considerably lower in the case of fabricated data (as observed in Fig. 12(b)). The use of fabricated data reduces greatly the averaged *mis* from 0.14160 to 0.02715 with the marginal change in averaged accuracy from 0.9771 to 0.9772. As an example, Fig. 13 shows the histograms of distances of training and test points from the affine hull of training samples (in Fig. 13(a)) and from the affine hull of fabricated samples (in Fig. 13(b)). The use of fabricated data in this example reduces the *mis* from 0.35919 to 0.05377 with loss of accuracy from 0.9858 to 0.9849.

6.2.2 Freiburg Groceries Dataset

The Freiburg groceries dataset is revisited to study the KAHM based differentially private classifiers. The 8192−dimensional feature vectors, extracted as stated in Section 6.1, are considered to study both differentially private classifier (Definition 12) and differentially private classifier based on fabricated data (Definition 15) for different values of privacy-loss bound ϵ and subspace dimension n while keeping the number of layers $L = 5$ and number of branches S as given in (59).

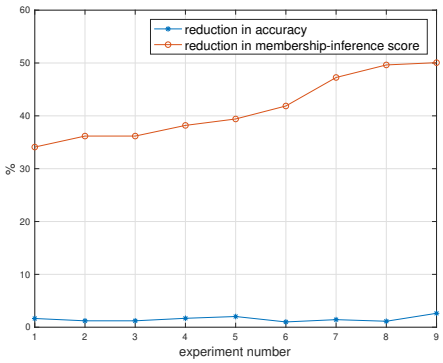
Table 6: Results of privacy-preserving learning experiments on Freiburg groceries dataset

(ϵ, n)	accuracy by classifier (Def. 15)	accuracy by classifier (Def. 12)	*mis* by classifier (Def. 15)	*mis* by classifier (Def. 12)
$(5, 20)$	0.8016	0.8153	0.02600	0.04207
$(8, 20)$	0.8556	0.8733	0.12512	0.20650
$(16, 20)$	0.8792	0.8919	0.27377	0.51909
$(32, 20)$	0.8811	0.8919	0.25753	0.40348
$(32, 5)$	0.7996	0.8212	0.04019	0.08048
$(32, 10)$	0.8595	0.8694	0.12220	0.24263
$(32, 15)$	0.8752	0.8841	0.19986	0.34380
$(32, 20)$	0.8811	0.8919	0.25753	0.40348
$(32, 25)$	0.8782	0.8929	0.29724	0.45099
	0.8568 (mean)	**0.8702** (mean)	**0.17772** (mean)	**0.29917** (mean)

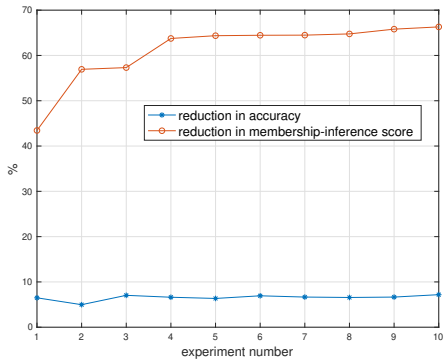
The experimental results are reported in Table 6. The averaged *msi* decreases from 0.29917 to 0.17772 together with the loss of averaged accuracy from 0.8702 to 0.8568. Fig. 14(a) illustrates the results via plotting the % reduction in both accuracy and *mis* values due to the use of fabricated data. It is observed from Fig. 14(a) that the use of fabricated data leads to a considerable reduction in *mis* value with relatively much smaller loss of accuracy. This is demonstrated through an example in Fig. 15 where the histograms of distances of training and test points from the affine hull of training samples (in Fig. 15(a)) and from the affine hull of fabricated samples (in Fig. 15(b)) are plotted. As the result of using fabricated data in this example, the *msi* reduces from 0.51320 to 0.12286 together with relatively smaller loss of accuracy from 0.8880 to 0.8615.

6.2.3 A Real Biomedical Dataset

A dataset related to the mental stress detection problem (Kumar et al., 2021, 2023) is considered to evaluate the proposed differentially private classifier based on fabricated data. This dataset consists of heart rate interval measurements of different subjects together with a stress-score on a scale from 0 to 100. The aim is to detect stress on an individual based on the analysis of recorded sequence of R-R intervals, $\{RR^i\}_i$. The R-R data vector at

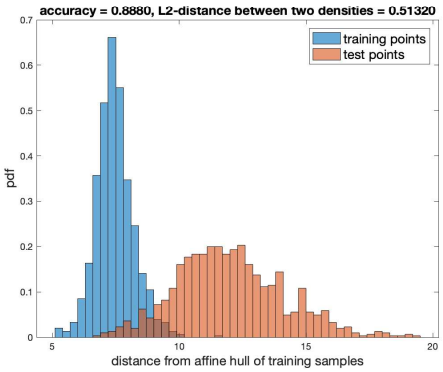


(a) Freiburg groceries dataset

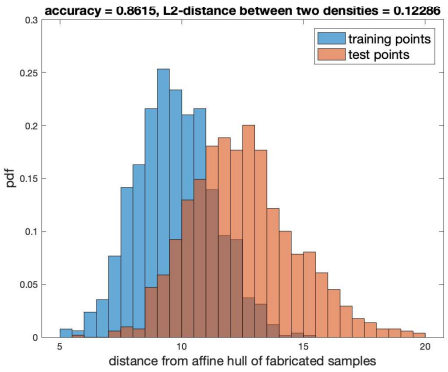


(b) heart rate variability dataset

Figure 14: The plots of % reduction in accuracy and *mis* values as a result of the use of fabricated data.



(a) histograms of distances of training and test data points from the affine hull of training samples



(b) histograms of distances of training and test data points from the affine hull of fabricated samples

Figure 15: Illustration of the reduction in *mis* value through data fabrication on Freiburg groceries dataset.

$i-$th time-index, y^i, is defined as $y^i = [\, RR^i \;\; RR^{i-1} \;\; \cdots \;\; RR^{i-d} \,]^T$. That is, the current interval and history of previous d intervals constitute the data vector. Assuming an average heartbeat of 72 beats per minute, d is chosen as equal to $72 \times 3 = 216$ so that R-R data vector consists of on an average 3-minutes long R-R intervals sequence. Following (Kumar et al., 2021, 2023), a dataset, say $\{y^i\}_i$, is built via 1) preprocessing the R-R interval sequence $\{RR^i\}_i$ with an impulse rejection filter for artifacts detection, and 2) excluding the R-R data vectors containing artifacts from the dataset. A class-label of either "*no-stress*" or "*under-stress*" is assigned to each data sample y^i based on the stress-score. For each subject, 50% of the data samples serve as training data while remaining as test data. Both differentially private classifier (Definition 12) and differentially private classifier based on fabricated data (Definition 15) are considered for the stress detection problem for different values of privacy-loss bound ϵ and subspace dimension n while keeping the number of layers $L = 5$ and number of branches S as given in (59).

Table 7: Results of privacy-preserving learning experiments on a biomedical dataset

(ϵ, n)	accuracy by classifier (Def. 15)	accuracy by classifier (Def. 12)	*mis* by classifier (Def. 15)	*mis* by classifier (Def. 12)
$(8, 5)$	0.8485	0.9074	0.02773	0.04903
$(16, 5)$	0.8817	0.9484	0.06739	0.15790
$(24, 5)$	0.8925	0.9557	0.07625	0.21036
$(32, 5)$	0.8964	0.9572	0.08310	0.23319
$(32, 1)$	0.8608	0.9058	0.00973	0.02259
$(32, 3)$	0.8777	0.9457	0.04045	0.12006
$(32, 5)$	0.8931	0.9568	0.07603	0.22235
$(32, 7)$	0.8960	0.9599	0.11834	0.33322
$(32, 10)$	0.8938	0.9566	0.17230	0.48887
$(32, 15)$	0.8795	0.9451	0.24173	0.68009
	0.8820 (mean)	**0.9439** (mean)	**0.09131** (mean)	**0.25177** (mean)

Table 7 reports the experimental results. The experimental results have been visualized in Fig. 14(b) via plotting the % reduction in both accuracy and *mis* values due to the use of fabricated data. It is observed from Fig. 14(b) that the use of fabricated data reduces considerably the *mis* value with relatively much smaller loss of accuracy. As a result of using fabricated data, the averaged *mis* decreases by 63.7328% (from 0.25177 to 0.09131 in absolute terms) together with averaged accuracy loss of 6.5579% (from 0.9439 to 0.8820 in absolute terms).

Table 8: Results of federated learning experiments under Scenario 1 on MNIST dataset

privacy-loss bound ϵ	accuracy by classifier (104) (distributed data)	accuracy by classifier (100) (centralized data)	change in accuracy due to data being distributed
1	0.9460	0.9478	-0.0018
1.5	0.9633	0.9629	0.0004
2	0.9710	0.9707	0.0003
3	0.9783	0.9779	0.0004
4	0.9790	0.9798	-0.0008
5	0.9813	0.9819	-0.0006
8	0.9829	0.9837	-0.0008
16	0.9847	0.9851	-0.0004
	0.9733 (mean)	**0.9737** (mean)	**-0.0004** (mean)

6.3 Federated Learning

MNIST dataset (containing the samples of 10 classes) is reconsidered under the following federated learning scenarios: Scenario 1: The training data are distributed among 10 parties such that all samples of a class are possessed by only a single party. That is, a party has all the samples of a class. Scenario 2: The training data are distributed among 20 parties such that samples of a class are shared equally between two parties. That is, a party has 50% samples of a class. Scenario 3: The training data are distributed randomly independent of the classes among Q number of parties where $Q \in \{2, 5, 10, 20, 50, 100\}$. For all of the considered scenarios, the local classifiers are built for privacy-loss bound $\epsilon \in \{1, 1, 5, 2, 3, 4, 5, 8, 16\}$, subspace dimension $n = 20$, number of layers $L = 5$, and number of branches S as given in (59). The performance of the global classifier (104) is evaluated on test data. As a reference, the performance in the case of non-federated learning (i.e. in the case of centralized data) is also evaluated using classifier (100).

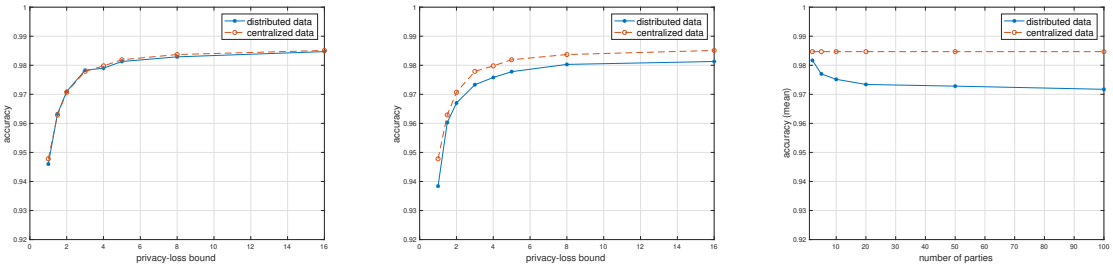
The obtained results are reported in Table 8 and Fig. 16(a) for Scenario 1, in Table 9 and Fig. 16(b) for Scenario 2, and in Table 10 and Fig. 16(c) for Scenario 3. Following observations are made from the obtained results: 1) In Scenario 1 (when samples of a class are not shared by parties), the federated learning performance is not different from that of learning with centralized data. This is expected, as for each class there exists only one local KAHM that serves as the global KAHM for that class as well. Since there remains no difference between the class specific global and local KAHMs, the performance in the federated setting remains unaffected. Thus, the change in accuracy due to data being distributed, as reported in Table 8, remains less than 0.0018. 2) In Scenario 2 (when samples of a class are shared by two parties), the performance under federated setting reduces slightly across the whole range of privacy-loss bound. It is observed from Table 9 that the change in accuracy due to data being distributed, averaged over the considered range of privacy-loss

Table 9: Results of federated learning experiments under Scenario 2 on MNIST dataset

privacy-loss bound ϵ	accuracy by classifier (104) (distributed data)	accuracy by classifier (100) (centralized data)	change in accuracy due to data being distributed
1	0.9384	0.9478	-0.0094
1.5	0.9603	0.9629	-0.0026
2	0.9670	0.9707	-0.0037
3	0.9733	0.9779	-0.0046
4	0.9758	0.9798	-0.0040
5	0.9778	0.9819	-0.0041
8	0.9803	0.9837	-0.0034
16	0.9813	0.9851	-0.0038
	0.9693 (mean)	**0.9737** (mean)	**-0.0044** (mean)

Table 10: Results of 10 independent federated learning experiments under Scenario 3 on MNIST dataset for privacy-loss bound $\epsilon = 16$

number of parties Q	mean accuracy by classifier (104) (distributed data)	accuracy by classifier (100) (centralized data)	change in mean accuracy due to data being distributed
2	0.9817	0.9847	-0.0030
5	0.9770	0.9847	-0.0077
10	0.9752	0.9847	-0.0095
20	0.9734	0.9847	-0.0113
50	0.9728	0.9847	-0.0119
100	0.9717	0.9847	-0.0130
	0.9753 (mean)	**0.9847** (mean)	**-0.0094** (mean)



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 16: The plots for the results of federated learning experiments on MNIST dataset.

bound, is equal to -0.0044. The accuracy loss of 0.0044 on an average due to distributed data is marginal indicating a good performance. 3) In Scenario 3 (when samples of a class are shared by up to 100 parties), it is observed from Table 10 that the decrease in performance due to data being distributed is not significantly high. Specifically, the data distribution among 100 parties did not cause a loss in accuracy of more than 0.013. This verifies the application potential of the proposed federated learning scheme.

6.4 Computational Time

We study the effect of sample size N only up to 1000 on the computational time of KAHM, as samples more than 1000 are divided into subsets and processed in parallel, as explained in Remark 7. Further, the effects of data dimension p and subspace dimension n are studied on the computational time of KAHM. For this, MATLAB R2017b simulations have been made on a MacBook Pro machine with a 2.2 GHz Intel Core i7 processor and 16 GB of memory. The simulations are made on the randomly generated data from the Gaussian distribution with mean 0 and variance 1 with

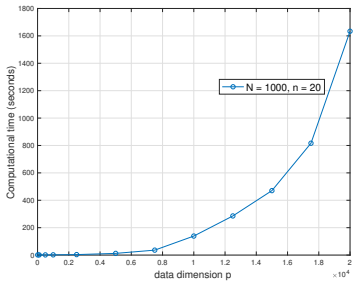
$$p \in \{10, 100, 500, 1000, 2500, 5000, 7500, 10000, 12500, 15000, 17500, 20000\}, \quad (105)$$

$$n \in \{5, 10, 20, 50, 75, 100, 200, 300, 400, 500\}, \text{ and} \quad (106)$$

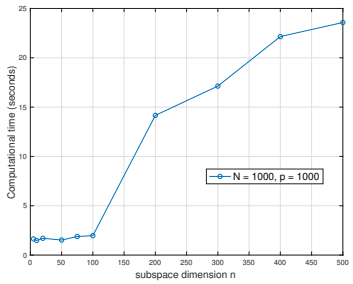
$$N \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}. \quad (107)$$

Fig. 17 plots the computational time of KAHM in relation to increasing data dimension p, subspace dimension n, sample size N, and data size $N \times p$. The results verify that KAHM remains computationally practical for a wide range of these parameters. Since the data dimension could be very high, simulations include the range of data dimension up to 20000. The results verify that 1) a higher data dimension does not pose a major computational challenge as it took around 1600 seconds to compute a KAHM from 1000 samples of 20000-dimensional data points, and 2) KAHM is computationally practical as observed from Table 11 that it took around 133 seconds to process a dataset with 10^7 entries (i.e. $N \times p = 10^7$).

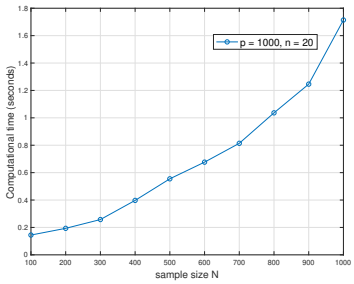
Remark 7 (Dealing with Large Data)**.** *To deal with the large data when the number of data samples N is large (say $N > 1000$), the data points are suggested to be divided into S number*



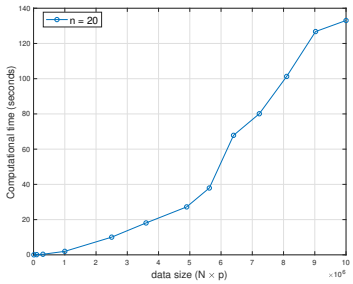
(a) against data dimension p



(b) against subspace dimension n



(c) against number of samples N



(d) against data size $N \times p$

Figure 17: The plots for the computational time of KAHM.

Table 11: Time required by KAHM to process data of varying sizes.

data size ($N \times p$)	processing time (in seconds)
10000	0.0193
100000	0.1077
300000	0.3232
1000000	1.9783
2500000	10.0412
3600000	18.1033
4900000	27.2175
5625000	37.9418
6400000	67.8467
7225000	80.1050
8100000	101.2839
9025000	126.7630
10000000	133.0497

of subsets (where S is given as in (59)) and for each subset a separate conditionally deep KAHM is built leading to a wide conditionally deep KAHM (Definition 5). Since each data subset can be processed independently, a large number of data samples can be computationally managed using parallel and distributed computing.

6.5 Summary of Experimental Results

Following inferences are drawn from the experiments:

1. KAHM based classifier improves the existing results on Freiburg groceries dataset indicating a highly competitive performance in modeling and thus classifying 8192-dimensional data points by means of KAHMs.

2. The proposed fabricated data based differentially private classification reduces considerably the risk of membership inference attack with relatively much smaller loss of accuracy. This is demonstrated through experiments on three different dataset: MNIST dataset, Freiburg groceries dataset, and a real biomedical dataset.

3. The application potential of the KAHM based differentially private federated learning scheme is verified by the observation that the accuracy-loss due to data being distributed is either marginal or not significantly high.

4. The computational issues arising from a large number of data samples are addressed automatically by design via splitting the dataset into subsets and processing each subset independently by a branch of the wide conditionally deep KAHM. Simulations verify that KAHM remains computationally practical and a higher data dimension does not pose a major computational challenge.

7. Conclusion

Having learned the representation of data samples in RKHS via solving a kernel regularized least squares problem with a meaningful choice of regularization parameter, KAHM defines a bounded geometric structure in the affine hull of data samples. KAHM and KAHM based models (consisting of series and parallel compositions of KAHMs) induce a distance function that measures the distance of an arbitrary data point from the data samples. Modeling the region of each class in data space through a separate KAHM allows building a classifier. An optimal differentially private noise adding mechanism is applied on training data samples to build a differentially private classifier. The smoothing of noise added samples through a carefully defined transformation (that ensures reducing the geometric modeling error of smoothed samples below of that of original samples) mitigates the accuracy-loss issue of the differentially private classifier.

The theoretical results obtained in this study are related to the determination of regularization parameter for the kernel regularized least squares problem, boundedness of KAHM, distance functions induced by KAHMs, and smoothing of data for reduction in KAHM modeling error. KAHMs can be applied to a wide range of machine learning problems and this study has considered the differentially private federated learning problem as an application example. The practical significance of the theory is demonstrated through numerous experiments performed to verify the application potential of the KAHMs. A significant feature

of our approach is that mathematical analysis has been carried out in a pure deterministic setting without making any statistical assumption.

Our future work will extend the KAHMs in several directions:

- A limitation of the current study is that the kernel function has been priori fixed of Gaussian type in defining the KAHM. The effect of different kernel functions and spectral properties of kernel matrix on the resulting geometric structures has not been investigated, which is the part of our future work.
- An advantage offered by the proposed approach is that it leverages the post-processing property of differential privacy and thus, unlike stochastic gradient descent based learning algorithms, there is no need of keeping track of the privacy loss incurred by successive iterations of an algorithm. However, in future we will also study the KAHM based iterative differentially private algorithms.
- KAHM approach will be extended to include a feature extraction procedure for the images, allowing for KAHMs to serve as a competitive alternative to the CNNs and a testing on large-scaled image datasets for a comparison with the existing models.
- Finally, the potential of KAHMs as deep generative models will be investigated.

Acknowledgments

The research reported in this paper has been supported by the Austrian Research Pro- motion Agency (FFG) COMET-Modul S3AI (Security and Safety for Shared Artificial Intelligence); FFG Grant SMiLe (Secure Machine Learning Applications with Homomorphically Encrypted Data); FFG Grant PRIMAL (Privacy Preserving Machine Learning for Industrial Applications); FFG Sub-Project PETAI (Privacy Secured Explainable and Transferable AI for Healthcare Systems); and the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry for Digital and Economic Affairs, and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE [(FFG grant no. 892418)] part of the FFG COMET Competence Centers for Excellent Technologies Programme.

Appendix A. Proof of Theorem 1

The proof is split into four parts.

Part 1: Consider

$$(Y)_{:,j} - K_X \left(K_X + (e+\tau)I_N\right)^{-1} (Y)_{:,j} = (I_N + \frac{1}{(e+\tau)}K_X)^{-1}(Y)_{:,j} \tag{108}$$

and thus

$$\mathcal{R}_{k,X,Y}(e,\tau) = \frac{1}{pN}\sum_{j=1}^{p}((Y)_{:,j})^T(I_N + \frac{1}{(e+\tau)}K_X)^{-2}(Y)_{:,j} \tag{109}$$

Since K_X is a positive definite matrix and $(e+\tau) > 0$,

$$\mu_{min}\left(I_N + \frac{1}{(e+\tau)}K_X\right) > 1 \tag{110}$$

where "$\mu_{min}(\cdot)$" denotes the minimum eigenvalue. Thus,

$$\mu_{max}\left(\left(I_N + \frac{1}{(e+\tau)}K_X\right)^{-2}\right) < 1 \tag{111}$$

where "$\mu_{max}(\cdot)$" denotes the maximum eigenvalue. This results in

$$\mathcal{R}_{k,X,Y}(e,\tau) < \frac{1}{pN}\sum_{j=1}^{p}\|(Y)_{:,j}\|^2 \tag{112}$$

$$= \frac{1}{pN}\|Y\|_F^2. \tag{113}$$

It is obvious that $\mathcal{R}_{k,X,Y}(e,\tau) > 0$, hence (22) follows.

Part 2: The derivative of $\mathcal{R}_{k,X,Y}$ w.r.t. e is given as

$$\frac{\mathrm{d}\mathcal{R}_{k,X,Y}(e,\tau)}{\mathrm{d}e} = \frac{2}{pN}\sum_{j=1}^{p}\Big\{(e+\tau)((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-2}(Y)_{:,j} \tag{114}$$

$$-(e+\tau)^2((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-3}(Y)_{:,j}\Big\}. \tag{115}$$

Consider

$$(e+\tau)^2((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-3}(Y)_{:,j} \tag{116}$$

$$\leq (e+\tau)^2\left\|\left((e+\tau)I_N + K_X\right)^{-1}\right\|_2((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-2}(Y)_{:,j} \tag{117}$$

$$= (e+\tau)\left\|\left(I_N + \frac{1}{(e+\tau)}K_X\right)^{-1}\right\|_2((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-2}(Y)_{:,j} \tag{118}$$

$$= (e+\tau)\frac{1}{\sigma_{min}\left(I_N + \frac{1}{(e+\tau)}K_X\right)}((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-2}(Y)_{:,j} \tag{119}$$

where "$\sigma_{min}(\cdot)$" denotes the minimum singular value. Observing that $(e+\tau) > 0$ and K_X is a positive definite matrix, we have

$$\sigma_{min}\left(I_N + \frac{1}{(e+\tau)}K_X\right) = 1 + \sigma_{min}\left(\frac{1}{(e+\tau)}K_X\right) \tag{120}$$

$$> 1. \tag{121}$$

Thus,

$$(e+\tau)^2((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-3}(Y)_{:,j}$$
$$< (e+\tau)((Y)_{:,j})^T\left((e+\tau)I_N + K_X\right)^{-2}(Y)_{:,j} \tag{122}$$

resulting in

$$\frac{\mathrm{d}\mathcal{R}_{k,X,Y}(e)}{\mathrm{d}e} > 0. \tag{123}$$

Since $(e+\tau) > 0$ and K_X is a positive definite matrix,

$$(e+\tau)^2((Y)_{:,j})^T \left((e+\tau)I_N + K_X\right)^{-3} (Y)_{:,j} > 0, \tag{124}$$

and therefore

$$\frac{\mathrm{d}\mathcal{R}_{k,X,Y}(e,\tau)}{\mathrm{d}e} < \frac{2}{pN}(e+\tau)\sum_{j=1}^{p}((Y)_{:,j})^T \left((e+\tau)I_N + K_X\right)^{-2} (Y)_{:,j} \tag{125}$$

$$= \frac{2}{(e+\tau)}\mathcal{R}_{k,X,Y}(e) \tag{126}$$

$$< \frac{2}{(e+\tau)}\frac{1}{pN}\|Y\|_F^2. \tag{127}$$

Inequalities (123) and (127) lead to (23).

Part 3: For a given $\tau \in \mathbf{R}_+$, introduce $m_\tau(e) = \mathcal{R}_{k,X,Y}(e,\tau) - e$, and observe that $m_\tau(0) > 0$ and $m_\tau(\frac{1}{pN}\|Y\|_F^2) < 0$. By the intermediate value theorem, there is a $\hat{e} \in (0, \frac{1}{pN}\|Y\|_F^2)$ such that $m_\tau(\hat{e}) = 0$, i.e., $\hat{e} = \mathcal{R}_{k,X,Y}(\hat{e},\tau)$. Thus, $\hat{e}$ is a fixed point of $\mathcal{R}_{k,X,Y}(e,\tau)$.

Part 4: It follows from (24) and (23) that

$$\frac{\mathrm{d}\mathcal{R}_{k,X,Y}(e,\tau)}{\mathrm{d}e} \in (0,1). \tag{128}$$

That is, there exists a constant c such that

$$0 < \frac{\mathrm{d}\mathcal{R}_{k,X,Y}(e|_{it},\tau)}{\mathrm{d}e} \leq c < 1,\ \forall it \in \{0,1,2,\cdots\}. \tag{129}$$

Let $\hat{e}$ be a fixed point of $\mathcal{R}_{k,X,Y}(e,\tau)$. Now, consider

$$|e|_{it} - \hat{e}| = |\mathcal{R}_{k,X,Y}(e|_{it-1},\tau) - \mathcal{R}_{k,X,Y}(\hat{e},\tau)| \leq c\,|e|_{it-1} - \hat{e}| \tag{130}$$

$$\leq c^2\,|e|_{it-2} - \hat{e}| \tag{131}$$

$$\vdots$$

$$\leq c^{it}\,|e|_0 - \hat{e}|\,, \tag{132}$$

that leads to

$$\lim_{it\to\infty} |e|_{it} - \hat{e}| \leq \lim_{it\to\infty} c^{it}\,|e|_0 - \hat{e}| = 0. \tag{133}$$

Hence the iterations (25)-(26) converge to a fixed point of $\mathcal{R}_{k,X,Y}(e,\tau)$. The uniqueness of the fixed point can be seen via assuming by contradiction that there exists another fixed point, say $\tilde{e}$. Now consider

$$|\tilde{e} - \hat{e}| = |\mathcal{R}_{k,X,Y}(\tilde{e},\tau) - \mathcal{R}_{k,X,Y}(\hat{e},\tau)| \leq c\,|\tilde{e} - \hat{e}| < |\tilde{e} - \hat{e}|\,. \tag{134}$$

This implies that $\tilde{e} = \hat{e}$. Hence, the result follows.

Appendix B. Proof of Theorem 2

Define a diagonal matrix D_y as

$$D_y = \text{diag}\left(k_\theta(Py, Py^1), \cdots, k_\theta(Py, Py^N)\right), \text{ and} \tag{135}$$

$$\overline{k}_y = \max_{i\in\{1,2,\cdots,N\}} k_\theta(Py, Py^i). \tag{136}$$

Further define a vector G_y as

$$G_y = \left[\, k_\theta(Py, Py^1) \;\cdots\; k_\theta(Py, Py^N) \,\right]^T. \tag{137}$$

It is obvious that $D_y^{-1} - (\overline{k}_y)^{-1} I$ is symmetric positive semi-definite, i.e.,

$$D_y^{-1} - (\overline{k}_y)^{-1} I_N \succeq 0. \tag{138}$$

Since K_{YP^T} is symmetric positive definite and $\lambda^* > 0$,

$$(K_{YP^T} + \lambda^* I_N)^{-1} \succ 0. \tag{139}$$

It follows from (138) and (139) that

$$\left(D_y^{-1} - (\overline{k}_y)^{-1} I_N\right)(K_{YP^T} + \lambda^* I_N)^{-1} \succeq 0, \text{ i.e.} \tag{140}$$

$$D_y^{-1}(K_{YP^T} + \lambda^* I_N)^{-1} - (\overline{k}_y)^{-1}(K_{YP^T} + \lambda^* I_N)^{-1} \succeq 0, \text{ i.e.} \tag{141}$$

$$G_y^T \left(D_y^{-1}(K_{YP^T} + \lambda^* I_N)^{-1} - (\overline{k}_y)^{-1}(K_{YP^T} + \lambda^* I_N)^{-1}\right) G_y \geq 0. \tag{142}$$

Thus,

$$G_y^T D_y^{-1}(K_{YP^T} + \lambda^* I_N)^{-1} G_y \geq (\overline{k}_y)^{-1} G_y^T (K_{YP^T} + \lambda^* I_N)^{-1} G_y. \tag{143}$$

Also,

$$\sum_{i=1}^{N} h^i_{k_\theta, YP^T, \lambda^*}(Py) = (\mathbf{1}_N)^T \left(K_{YP^T} + \lambda I_N\right)^{-1} G_y \tag{144}$$

$$= G_y^T D_y^{-1} \left(K_{YP^T} + \lambda I_N\right)^{-1} G_y \tag{145}$$

$$\geq (\overline{k}_y)^{-1} G_y^T (K_{YP^T} + \lambda^* I_N)^{-1} G_y \tag{146}$$

$$\geq (\overline{k}_y)^{-1} \mu_{min}\left((K_{YP^T} + \lambda^* I_N)^{-1}\right) \|G_y\|^2. \tag{147}$$

As $(K_{YP^T} + \lambda^* I_N)^{-1}$ is real symmetric positive definite,

$$\sum_{i=1}^{N} h^i_{k_\theta, YP^T, \lambda^*}(Py) > 0. \tag{148}$$

Consider

$$\frac{\left\| \left[\, h^1_{k_\theta, YP^T, \lambda^*}(Py) \;\cdots\; h^N_{k_\theta, YP^T, \lambda^*}(Py) \,\right]^T \right\|}{\left| \sum_{i=1}^{N} h^i_{k_\theta, YP^T, \lambda^*}(Py) \right|} = \frac{\left\| (K_{YP^T} + \lambda^* I_N)^{-1} G_y \right\|}{\sum_{i=1}^{N} h^i_{k_\theta, YP^T, \lambda^*}(Py)} \tag{149}$$

$$\leq \frac{\overline{k}_y}{\|G_y\|} \frac{\left\| (K_{YP^T} + \lambda^* I_N)^{-1} \right\|_2}{\mu_{min}\left((K_{YP^T} + \lambda^* I_N)^{-1}\right)}. \tag{150}$$

Since $(K_{YP^T} + \lambda^* I_N)^{-1}$ is real symmetric positive definite,

$$\frac{\overline{k}_y}{\|G_y\|} \frac{\left\|(K_{YP^T} + \lambda^* I_N)^{-1}\right\|_2}{\mu_{min}\left((K_{YP^T} + \lambda^* I_N)^{-1}\right)} = \frac{\overline{k}_y}{\|G_y\|} \frac{\mu_{max}\left((K_{YP^T} + \lambda^* I_N)^{-1}\right)}{\mu_{min}\left((K_{YP^T} + \lambda^* I_N)^{-1}\right)} \tag{151}$$

$$= \frac{\overline{k}_y}{\|G_y\|} \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} \tag{152}$$

$$= \frac{\max_{i\in\{1,2,\cdots,N\}}\ k_\theta(Py, Py^i)}{\sqrt{\sum_{i=1}^N |k_\theta(Py, Py^i)|^2}} \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} \tag{153}$$

$$< \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})}. \tag{154}$$

Thus

$$\frac{\left\|\left[\, h^1_{k_\theta,YP^T,\lambda^*}(Py)\ \cdots\ h^N_{k_\theta,YP^T,\lambda^*}(Py)\,\right]^T\right\|}{\left|\sum_{i=1}^N h^i_{k_\theta,YP^T,\lambda^*}(Py)\right|} < \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})}. \tag{155}$$

Since K_{YP^T} is positive definite (i.e. $\mu_{min}(K_{YP^T}) > 0$) and $\mathrm{tr}(K_{YP^T}) = N$ (i.e. $\mu_{max}(K_{YP^T}) < N$), we have

$$\frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} < \frac{\lambda^* + N}{\lambda^*}. \tag{156}$$

Using (40) with the observation that $\hat{e} > 0$, we have

$$\lambda^* > \frac{2}{pN}\|Y\|_F^2, \text{ leading to} \tag{157}$$

$$\frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} < 1 + \frac{pN^2}{2\|Y\|_F^2}. \tag{158}$$

It is observed from (33) that

$$\mathcal{A}_{Y,n}(y) = \frac{1}{\sum_{i=1}^N h^i_{k_\theta,YP^T,\lambda^*}(Py)} [\, y^1\ \cdots\ y^N\,] \left[\, h^1_{k_\theta,YP^T,\lambda^*}(Py)\ \cdots\ h^N_{k_\theta,YP^T,\lambda^*}(Py)\,\right]^T. \tag{159}$$

Thus,

$$\|\mathcal{A}_{Y,n}(y)\| \le \|[\, y^1\ \cdots\ y^N\,]\|_2 \frac{\left\|\left[\, h^1_{k_\theta,YP^T,\lambda^*}(Py)\ \cdots\ h^N_{k_\theta,YP^T,\lambda^*}(Py)\,\right]^T\right\|}{\left|\sum_{i=1}^N h^i_{k_\theta,YP^T,\lambda^*}(Py)\right|}. \tag{160}$$

Using (155) and (158) in (160) leads to

$$\|\mathcal{A}_{Y,n}(y)\| < \|Y\|_2 \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} < \|Y\|_2 \left(1 + \frac{pN^2}{2\|Y\|_F^2}\right). \tag{161}$$

Hence, (46) follows.

Appendix C. Proof of Theorem 3

It is observed from (33) that

$$y - \mathcal{A}_{Y,n}(y) = \frac{1}{\sum_{i=1}^{N} h^i_{k_\theta,YP^T,\lambda^*}(Py)} [\, y - y^1 \ \cdots \ y - y^N \,] \left[h^1_{k_\theta,YP^T,\lambda^*}(Py) \ \cdots \ h^N_{k_\theta,YP^T,\lambda^*}(Py) \right]^T . \quad (162)$$

Thus,

$$\|y - \mathcal{A}_{Y,n}(y)\| \leq \|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2 \frac{\left\| \left[h^1_{k_\theta,YP^T,\lambda^*}(Py) \ \cdots \ h^N_{k_\theta,YP^T,\lambda^*}(Py) \right]^T \right\|}{\left| \sum_{i=1}^{N} h^i_{k_\theta,YP^T,\lambda^*}(Py) \right|}. \quad (163)$$

That is,

$$\frac{\Gamma_{\mathcal{A}_{Y,n}}(y)}{\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2} \leq \frac{\left\| \left[h^1_{k_\theta,YP^T,\lambda^*}(Py) \ \cdots \ h^N_{k_\theta,YP^T,\lambda^*}(Py) \right]^T \right\|}{\left| \sum_{i=1}^{N} h^i_{k_\theta,YP^T,\lambda^*}(Py) \right|}. \quad (164)$$

Using (155) and (158) leads to

$$\frac{\Gamma_{\mathcal{A}_{Y,n}}(y)}{\|[\, y - y^1 \ \cdots \ y - y^N \,]\|_2} < \frac{\lambda^* + \mu_{max}(K_{YP^T})}{\lambda^* + \mu_{min}(K_{YP^T})} < 1 + \frac{pN^2}{2\|Y\|_F^2}. \quad (165)$$

Hence, (49) follows.

Appendix D. Proof of Theorem 4

It is observed from the definition of $\mathcal{D}_{Y,n,L}$ (i.e. (50-52)) that

$$\Gamma_{\mathcal{D}_{Y,n,L}}(y) \leq \Gamma_{\mathcal{A}_{Y,n}}(y). \quad (166)$$

Using (166) in (49) leads to (54).

Appendix E. Proof of Theorem 5

$$\begin{aligned}
\Gamma_{\mathcal{W}_{Y,n,L,S}}(y) &= \min_{s \in \{1,2,\cdots,S\}} \Gamma_{\mathcal{D}_{Y_s,n,L}}(y) && (167)\\
&\leq \min_{s \in \{1,2,\cdots,S\}} \Gamma_{\mathcal{A}_{Y_s,n}}(y) && (168)\\
&< \min_{s \in \{1,2,\cdots,S\}} \left\{ \left(1 + \frac{pN_s^2}{2\|Y_s\|_F^2} \right) \|[\, y - y^{1,s} \ \cdots \ y - y^{N_s,s} \,]\|_2 \right\} && (169)\\
&\leq \min_{s \in \{1,2,\cdots,S\}} \left\{ \left(1 + \frac{pN_s^2}{2\|Y_s\|_F^2} \right) \|[\, y - y^{1,s} \ \cdots \ y - y^{N_s,s} \,]\|_F \right\}. && (170)
\end{aligned}$$

Since $\{y^{1,s}, \cdots, y^{N_s,s}\} \subset \{y^1, \cdots, y^N\}$,

$$\|[\, y - y^{1,s} \ \cdots \ y - y^{N_s,s} \,]\|_F < \|[\, y - y^1 \ \cdots \ y - y^N \,]\|_F , \quad (171)$$

and thus

$$\Gamma_{\mathcal{W}_{Y,n,L,S}}(y) < \|[\, y - y^1 \ \cdots \ y - y^N \,]\|_F \times \min_{s\in\{1,2,\cdots,S\}} \left(1 + \frac{pN_s^2}{2\|Y_s\|_F^2}\right) \tag{172}$$

leading to (61).

Appendix F. Proof of Theorem 6

Define a $N \times N$ matrix H_m and a $p \times N$ matrix O_m as

$$H_m = \begin{bmatrix} h^1_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{1,m}) & \cdots & h^1_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{N,m}) \\ \vdots & & \vdots \\ h^N_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{1,m}) & \cdots & h^N_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{N,m}) \end{bmatrix}, \tag{173}$$

$$O_m = \begin{bmatrix} \sum_{j=1}^N h^j_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{1,m}) & \cdots & \sum_{j=1}^N h^j_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{N,m}) \\ \vdots & & \vdots \\ \sum_{j=1}^N h^j_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{1,m}) & \cdots & \sum_{j=1}^N h^j_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{N,m}) \end{bmatrix}_{p\times N}. \tag{174}$$

It can be seen using (38) that

$$H_m = (K_{\hat{Y}_m P_m^T} + \lambda_m^* I_N)^{-1} K_{\hat{Y}_m P_m^T}, \text{ i.e.} \tag{175}$$

$$I_N - H_m = (I_N + \frac{1}{\lambda_m^*} K_{\hat{Y}_m P_m^T})^{-1}, \text{ i.e.} \tag{176}$$

$$\|I_N - H_m\|_2 = \frac{1}{\sigma_{min}(I_N + \frac{1}{\lambda_m^*} K_{\hat{Y}_m P_m^T})}. \tag{177}$$

As $\lambda_m^* > 0$ and $K_{\hat{Y}_m P_m^T}$ is a positive definite matrix,

$$\|I_N - H_m\|_2 = \frac{1}{1 + \sigma_{min}(\frac{1}{\lambda_m^*} K_{\hat{Y}_m P_m^T})} < 1. \tag{178}$$

As the r.h.s. of (176) is a positive definite matrix, we have

$$\mu_{min}(I_N - H_m) > 0, \text{ i.e.} \tag{179}$$

$$\mu_{max}(H_m) < 1. \tag{180}$$

As H_m is a real symmetric matrix, it follows immediately from (180) that $\sigma_{max}(H_m) < 1$, and thus

$$\|H_m\|_2 < 1, \text{ i.e.} \tag{181}$$

$$\|H_m\|_1 < \sqrt{N}, \text{ i.e.} \tag{182}$$

$$\max_{i\in\{1,2,\cdots,N\}} \sum_{j=1}^N |h^j_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{i,m})| < \sqrt{N}, \text{ thus} \tag{183}$$

$$\sum_{j=1}^N h^j_{k_{\theta_m},\hat{Y}_m P_m^T,\lambda_m^*}(P_m\hat{y}^{i,m}) < \sqrt{N}, \ \forall i \in \{1,2,\cdots,N\}. \tag{184}$$

In view of (148), we have

$$0 < \sum_{j=1}^{N} h^{j}_{k_{\theta_m}, \hat{Y}_m P_m^T, \lambda_m^*}(P_m \hat{y}^{i,m}) < \sqrt{N},\ \forall i \in \{1, 2, \cdots, N\}. \tag{185}$$

It follows from (33) that

$$O_m \circ \left[\mathcal{A}_{\hat{Y}_m, n}(\hat{y}^{1,m}) \ \cdots \ \mathcal{A}_{\hat{Y}_m, n}(\hat{y}^{N,m}) \right] = [\, \hat{y}^{1,m} \ \cdots \ \hat{y}^{N,m} \,] H_m, \text{ i.e.} \tag{186}$$

$$\hat{Y}_{m+1}^T = \hat{Y}_m^T H_m, \text{ i.e.} \tag{187}$$

$$\hat{Y}_m^T - \hat{Y}_{m+1}^T = \hat{Y}_m^T (I_N - H_m). \tag{188}$$

Using (188) for $m = M-1$, we have

$$\hat{Y}_{M-1}^T - \hat{Y}_M^T = \hat{Y}_{M-1}^T (I_N - H_{M-1}). \tag{189}$$

Using (187) recursively from $m = 0$ to $m = M-2$, we have

$$\hat{Y}_{M-1}^T = \hat{Y}_0^T H_0 H_1 \cdots H_{M-2}. \tag{190}$$

Combining (189) and (190) leads to

$$\hat{Y}_{M-1}^T - \hat{Y}_M^T = \hat{Y}_0^T H_0 H_1 \cdots H_{M-2}(I_N - H_{M-1}), \text{ i.e.} \tag{191}$$

$$\hat{Y}_{M-1} - \hat{Y}_M = (I_N - H_{M-1}) H_{M-2} \cdots H_1 H_0 \hat{Y}_0, \text{ i.e.} \tag{192}$$

$$\|\hat{Y}_{M-1} - \hat{Y}_M\|_F \leq \|(I_N - H_{M-1}) H_{M-2} \cdots H_1 H_0\|_2 \|\hat{Y}_0\|_F, \text{ i.e.} \tag{193}$$

$$\|\hat{Y}_{M-1} - \hat{Y}_M\|_F \leq \|I_N - H_{M-1}\|_2 \|H_{M-2}\|_2 \cdots \|H_1\|_2 \|H_0\|_2 \|\hat{Y}_0\|_F. \tag{194}$$

Define

$$\beta = \max\ (\|I_N - H_{M-1}\|_2, \|H_{M-2}\|_2, \cdots, \|H_1\|_2, \|H_0\|_2)\,. \tag{195}$$

Since $\|H_m\|_2 < 1$ (i.e. (181)) and also $\|I_N - H_m\|_2 < 1$ (i.e. (178)), we must have

$$0 < \beta < 1. \tag{196}$$

It follows from (194) that

$$\|\hat{Y}_{M-1} - \hat{Y}_M\|_F \leq (\beta)^M \|\hat{Y}_0\|_F. \tag{197}$$

Considering that $\hat{Y}_0 = Y_\epsilon^+$ and $\hat{y}^{i,m}$ is the $i-$th column of $\hat{Y}_m^T$, we have

$$\|\hat{y}^{i,M-1} - \hat{y}^{i,M}\| \leq (\beta)^M \|Y_\epsilon^+\|_F. \tag{198}$$

Consider

$$\begin{aligned} &\hat{y}^{i,M} - \mathcal{A}_{\hat{Y}_{M-1}, n}\left(\hat{y}^{i,M-1}\right) \\ &= \left(\sum_{j=1}^{N} h^{j}_{k_{\theta_{M-1}}, \hat{Y}_{M-1} P_{M-1}^T, \lambda_{M-1}^*}(P_{M-1} \hat{y}^{i,M-1}) - 1 \right) \mathcal{A}_{\hat{Y}_{M-1}, n}\left(\hat{y}^{i,M-1}\right), \text{ thus} \end{aligned} \tag{199}$$

$$\|\hat{y}^{i,M} - \mathcal{A}_{\hat{Y}_{M-1},n}\left(\hat{y}^{i,M-1}\right)\| = \left|\sum_{j=1}^{N} h^{j}_{k_{\theta_{M-1}},\hat{Y}_{M-1}P_{M-1}^{T},\lambda^{*}_{M-1}}(P_{M-1}\hat{y}^{i,M-1}) - 1\right| \|\mathcal{A}_{\hat{Y}_{M-1},n}\left(\hat{y}^{i,M-1}\right)\|. \quad (200)$$

Using (185) in (200), we have

$$\|\hat{y}^{i,M} - \mathcal{A}_{\hat{Y}_{M-1},n}\left(\hat{y}^{i,M-1}\right)\| < (\sqrt{N}-1)\|\mathcal{A}_{\hat{Y}_{M-1},n}\left(\hat{y}^{i,M-1}\right)\|. \quad (201)$$

The inequality (46) leads to

$$\|\hat{y}^{i,M} - \mathcal{A}_{\hat{Y}_{M-1},n}\left(\hat{y}^{i,M-1}\right)\| < (\sqrt{N}-1)\|\hat{Y}_{M-1}\|_2 \frac{\lambda^{*}_{M-1} + \mu_{max}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}{\lambda^{*}_{M-1} + \mu_{min}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})} \quad (202)$$

$$< (\sqrt{N}-1)\|\hat{Y}_{M-1}\|_F \frac{\lambda^{*}_{M-1} + \mu_{max}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}{\lambda^{*}_{M-1} + \mu_{min}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}. \quad (203)$$

It follows from (190) that

$$\|\hat{Y}_{M-1}\|_F \leq \|H_{M-2}\|_2 \cdots \|H_1\|_2 \|H_0\|_2 \|\hat{Y}_0\|_F, \text{ i.e.} \quad (204)$$

$$\|\hat{Y}_{M-1}\|_F \leq (\beta)^{M-1}\|Y_{\epsilon}^{+}\|_F. \quad (205)$$

Thus

$$\|\hat{y}^{i,M} - \mathcal{A}_{\hat{Y}_{M-1},n}\left(\hat{y}^{i,M-1}\right)\| < (\sqrt{N}-1)(\beta)^{M-1} \frac{\lambda^{*}_{M-1} + \mu_{max}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}{\lambda^{*}_{M-1} + \mu_{min}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})} \|Y_{\epsilon}^{+}\|_F. \quad (206)$$

Consider

$$\|\hat{y}^{i,M-1} - \mathcal{A}_{\hat{Y}_{M-1},n}(\hat{y}^{i,M-1})\| \leq \|\hat{y}^{i,M-1} - \hat{y}^{i,M}\| + \|\hat{y}^{i,M} - \mathcal{A}_{\hat{Y}_{M-1},n}\left(\hat{y}^{i,M-1}\right)\|. \quad (207)$$

Using (198) and (206) in (207), we finally obtain

$$\|\hat{y}^{i,M-1} - \mathcal{A}_{\hat{Y}_{M-1},n}(\hat{y}^{i,M-1})\| < (\beta)^{M-1}\|Y^{+}\|_F \left(\beta + \frac{\lambda^{*}_{M-1} + \mu_{max}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}{\lambda^{*}_{M-1} + \mu_{min}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}(\sqrt{N}-1)\right). \quad (208)$$

Since $\mu_{min}(K_{\hat{Y}_{M-1}P_{M-1}^{T}}) > 0$, $\mu_{max}(K_{\hat{Y}_{M-1}P_{M-1}^{T}}) < N$, $\lambda^{*}_{M-1} > 0$, and $0 < \beta < 1$, we have

$$0 < \left(\beta + \frac{\lambda^{*}_{M-1} + \mu_{max}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}{\lambda^{*}_{M-1} + \mu_{min}(K_{\hat{Y}_{M-1}P_{M-1}^{T}})}(\sqrt{N}-1)\right) < 1 + \frac{\lambda^{*}_{M-1} + N}{\lambda^{*}_{M-1}}(\sqrt{N}-1). \quad (209)$$

Thus,

$$0 \leq \frac{\|\hat{y}^{i,M-1} - \mathcal{A}_{\hat{Y}_{M-1},n}(\hat{y}^{i,M-1})\|}{(\beta)^{M-1}} < \|Y^{+}\|_F \left(1 + \frac{\lambda^{*}_{M-1} + N}{\lambda^{*}_{M-1}}(\sqrt{N}-1)\right) < \infty \quad (210)$$

Further, it is observed from (196) that

$$\lim_{M\to\infty} (\beta)^{M-1} = 0, \text{ and thus} \tag{211}$$

$$\lim_{M\to\infty} \|\hat{y}^{i,M-1} - \mathcal{A}_{\hat{Y}_{M-1},n}(\hat{y}^{i,M-1})\| = 0. \tag{212}$$

Since (212) holds for all $i \in \{1, 2, \cdots, N\}$, we must have

$$\lim_{M\to\infty} \sum_{i=1}^{N} \|\hat{y}^{i,M-1} - \mathcal{A}_{\hat{Y}_{M-1},n}(\hat{y}^{i,M-1})\| = 0. \tag{213}$$

Hence, the result is proved.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA. ACM.

Balle, B., & Wang, Y. (2018). Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In Dy, J. G., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 403–412. PMLR.

Belkin, M., Ma, S., & Mandal, S. (2018). To understand deep learning we need to understand kernel learning. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 541–549. PMLR.

Cevikalp, H., Triggs, B., Yavuz, H. S., Küçük, Y., Küçük, M., & Barkana, A. (2010). Large margin classifiers based on affine hulls. *Neurocomputing*, *73*(16), 3160–3168.

Chaudhuri, K., Monteleoni, C., & Sarwate, A. (2011). Differentially private empirical risk minimization. *Journal of Machine Learning Research*, *12*, 1069–1109.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In Halevi, S., & Rabin, T. (Eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, *9*(3-4), 211–407.

Geng, Q., Kairouz, P., Oh, S., & Viswanath, P. (2015). The staircase mechanism in differential privacy. *IEEE Journal of Selected Topics in Signal Processing*, *9*(7), 1176–1184.

Geng, Q., & Viswanath, P. (2016a). The optimal noise-adding mechanism in differential privacy. *IEEE Transactions on Information Theory*, *62*(2), 925–951.

Geng, Q., & Viswanath, P. (2016b). Optimal noise adding mechanisms for approximate differential privacy. *IEEE Transactions on Information Theory*, *62*(2), 952–969.

Geng, Q., Ding, W., Guo, R., & Kumar, S. (2018). Optimal noise-adding mechanism in additive differential privacy. In *International Conference on Artificial Intelligence and Statistics*.

Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2021). Reproducing kernel hilbert space, mercer's theorem, eigenfunctions, nyström method, and use of kernels in machine learning: Tutorial and survey. *ArXiv*, *abs/2106.08443*.

Gholami, B., & Hajisami, A. (2016). Kernel auto-encoder for semi-supervised hashing. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–8.

Ghosh, A., Roughgarden, T., & Sundararajan, M. (2012). Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, *41*(6), 1673–1693.

Gupte, M., & Sundararajan, M. (2010). Universally optimal privacy mechanisms for minimax agents. In *Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '10, pp. 135–146, New York, NY, USA. ACM.

Hall, R., Rinaldo, A., & Wasserman, L. (2013). Differential privacy for functions and functional data. *J. Mach. Learn. Res.*, *14*(1), 703–727.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*(3), 1171 – 1220.

Jain, P., & Thakurta, A. (2013). Differentially private learning with kernels. In Dasgupta, S., & McAllester, D. (Eds.), *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28 of *Proceedings of Machine Learning Research*, pp. 118–126, Atlanta, Georgia, USA. PMLR.

Jund, P., Abdo, N., Eitel, A., & Burgard, W. (2016). The freiburg groceries dataset. *CoRR*, *abs/1611.05799*.

Kampffmeyer, M., Løkse, S., Bianchi, F. M., Jenssen, R., & Livi, L. (2018). The deep kernelized autoencoder. *Applied Soft Computing*, *71*, 816–825.

Kumar, M., Rossbory, M., Moser, B. A., & Freudenthaler, B. (2019). Deriving an optimal noise adding mechanism for privacy-preserving machine learning. In Anderst-Kotsis, G., Tjoa, A. M., Khalil, I., Elloumi, M., Mashkoor, A., Sametinger, J., Larrucea, X., Fensel, A., Martinez-Gil, J., Moser, B., Seifert, C., Stein, B., & Granitzer, M. (Eds.), *Proceedings of the 3rd International Workshop on Cyber-Security and Functional Safety in Cyber-Physical (IWCFS 2019), August 26-29, 2019, Linz, Austria*, pp. 108–118, Cham. Springer International Publishing.

Kumar, M., Rossbory, M., Moser, B. A., & Freudenthaler, B. (2021). An optimal $(\epsilon, \delta)-$differentially private learning of distributed deep fuzzy models. *Information Sciences*, *546*, 87–120.

Kumar, M. (2023). Differentially private transferrable deep learning with membership-mappings. *Advances in Computational Intelligence*, *3*(1), 1–27.

Kumar, M., & Freudenthaler, B. (2020). Fuzzy membership functional analysis for nonparametric deep models of image features. *IEEE Transactions on Fuzzy Systems*, *28*(12), 3345–3359.

Kumar, M., Moser, B., Fischer, L., & Freudenthaler, B. (2021a). Membership-mappings for data representation learning: A bregman divergence based conditionally deep autoencoder. In Kotsis, G., Tjoa, A. M., Khalil, I., Moser, B., Mashkoor, A., Sametinger, J., Fensel, A., Martinez-Gil, J., Fischer, L., Czech, G., Sobieczky, F., & Khan, S. (Eds.), *Database and Expert Systems Applications - DEXA 2021 Workshops*, pp. 138–147, Cham. Springer International Publishing.

Kumar, M., Moser, B., Fischer, L., & Freudenthaler, B. (2021b). Membership-mappings for data representation learning: Measure theoretic conceptualization. In Kotsis, G., Tjoa, A. M., Khalil, I., Moser, B., Mashkoor, A., Sametinger, J., Fensel, A., Martinez-Gil, J., Fischer, L., Czech, G., Sobieczky, F., & Khan, S. (Eds.), *Database and Expert Systems Applications - DEXA 2021 Workshops*, pp. 127–137, Cham. Springer International Publishing.

Kumar, M., Rossbory, M., Moser, B. A., & Freudenthaler, B. (2020). Differentially private learning of distributed deep models. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '20 Adjunct, pp. 193–200, New York, NY, USA. Association for Computing Machinery.

Kumar, M., Singh, S., & Freudenthaler, B. (2021). Gaussian fuzzy theoretic analysis for variational learning of nested compositions. *International Journal of Approximate Reasoning*, *131*, 1–29.

Kumar, M., Zhang, W., Fischer, L., & Freudenthaler, B. (2023). Membership-mappings for practical secure distributed deep learning..

Kumar, M., Zhang, W., Weippert, M., & Freudenthaler, B. (2021). An explainable fuzzy theoretic nonparametric deep model for stress assessment using heartbeat intervals analysis. *IEEE Transactions on Fuzzy Systems*, *29*(12), 3873–3886.

Laforgue, P., Clémençon, S., & d'Alché-Buc, F. (2019). Autoencoding any data through kernel autoencoders. In Chaudhuri, K., & Sugiyama, M. (Eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, Vol. 89 of *Proceedings of Machine Learning Research*, pp. 1061–1069. PMLR.

Nikhitha, N. K., Afzal, A. L., & Asharaf, S. (2021). Deep kernel machines: A survey. *Pattern Anal. Appl.*, *24*(2), 537–556.

Park, M., Foulds, J., Chaudhuri, K., & Welling, M. (2020). Variational bayes in private settings (vips). *Journal of Artificial Intelligence Research*, *68*, 109–157.

Phan, N., Wang, Y., Wu, X., & Dou, D. (2016). Differential privacy preservation for deep auto-encoders: An application of human behavior prediction. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 1309–1316. AAAI Press.

Rudi, A., Carratino, L., & Rosasco, L. (2017). Falkon: An optimal large scale kernel method. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

Schölkopf, B., Herbrich, R., & Smola, A. J. (2001). A generalized representer theorem. In Helmbold, D., & Williamson, B. (Eds.), *Computational Learning Theory*, pp. 416–426, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sugiyama, M., Kanamori, T., Suzuki, T., Plessis, M. C. d., Liu, S., & Takeuchi, I. (2013). Density-Difference Estimation. *Neural Computation*, *25*(10), 2734–2775.

Wilson, A. G., Hu, Z., Salakhutdinov, R., & Xing, E. P. (2016). Deep kernel learning. In Gretton, A., & Robert, C. C. (Eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, Vol. 51 of *Proceedings of Machine Learning Research*, pp. 370–378, Cadiz, Spain. PMLR.

Zhang, Q., Yang, J., Zhang, W., Kumar, M., Liu, J., Liu, J., & Li, X. (2023). Deep fuzzy mapping nonparametric model for real-time demand estimation in water distribution systems: A new perspective. *Water Research*, *241*, 120145.

Zhang, W., Kumar, M., Ding, W., Li, X., & Yu, J. (2022). Variational learning of deep fuzzy theoretic nonparametric model. *Neurocomputing*, *506*, 128–145.

Zhang, Y., Hao, Z., & Wang, S. (2019). A differential privacy support vector machine classifier based on dual variable perturbation. *IEEE Access*, *7*, 98238–98251.