Project no. 826278

SERUMS

Research & Innovation Action (RIA)
SECURING MEDICAL DATA IN SMART PATIENT-CENTRIC HEALTHCARE SYSTEMS

Report on Initial Distributed Differential Privacy Deep Learning Models
D3.1

Due date of deliverable: 31st December 2019

Start date of project: January 1st, 2019

Type: Deliverable
WP number: WP3

Responsible institution: Software Competence Center Hagenberg
Editor and editor's address: Michael Rossbory, Software Competence Center Hagenberg

Version 1.0

Project co-funded by the European Commission within the Horizon 2020 Programme		
Dissemination Level		
PU	Public	√
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Change Log

Rev.	Date	Who	Site	What
1	01/12/19	Michael Rossbory	SCCH	Document Structure
2	17/12/19	Mohit Kumar	SCCH	Initial Version

Executive Summary

This report introduces a privacy-preserving framework to distributed deep learning. Assuming training data as private, the problem of learning of local deep models is considered in a distributed setting under differential privacy framework. A local deep fuzzy model, formed by a composition of a finite number of Takagi-Sugeno type fuzzy filters, is learned using variational Bayesian inference. This study suggests an optimal $(\epsilon, \delta)-$differentially private noise adding mechanism that results in multi-fold reduction in noise magnitude over the classical Gaussian mechanism and thus leads to an increased utility for a given level of privacy. Further, the robustness feature, offered by the rule-based fuzzy systems, is leveraged to alleviate the effect of added data noise on the utility. An architecture for distributed form of differentially private learning is suggested where a privacy wall separates the private local training data from the globally shared data, and fuzzy sets and fuzzy rules are used to aggregate robustly the local deep fuzzy models for building the global model. The privacy wall uses noise adding mechanisms to attain differential privacy for each participant's private training data and thus the adversaries have no direct access to the training data.

Contents

1. Introduction

As the modern information technology enables acquisition and storage of increasingly detailed private data, an increasingly rising interest in a rigorous mathematical study of definition of privacy and the privacy preserving computational algorithms is natural. The goal is to simultaneously protect private data of individuals in a dataset while permitting statistical and computational analysis of the dataset as a whole. The anonymization of data might not preserve the privacy as a *linkage attack* [1] is possible against anonymized data where the richness of data can be exploited to match anonymized records with non-anonymized records in a different dataset. The re-identification of anonymized records might reveal a compromising information that could cause harm to an individual. The releasing summary statistics is also a privacy risk posed by *reconstruction attacks* [2]. Differential Privacy [3, 4] is a formalism dedicated to the problem of privacy-preserving data analysis. Differential privacy is a formal framework to quantify the degree to which the privacy for each individual in the dataset is preserved while releasing the output of a data analysis algorithm. Differential privacy guarantees that an adversary, by virtue of presence or absence of an individual's data in the dataset, can't draw any conclusions about an individual from the released output of the analysis algorithm. Differential privacy, being a property of an algorithm's data access mechanism, automatically neutralizes linkage attacks and provides protection against arbitrary risks. Further, differential privacy remains immune to any post-processing on the output of the private algorithm.

The datasets required for the learning of models might be containing sensitive information that need to be protected from *model inversion* attack [5] and from adversaries with access to model parameters and knowledge of the training procedure. This goal has been addressed within the framework of differential privacy [6, 7]. The classical approach for approximating a real-valued function with a differential private mechanism is to perturb the function output via adding noise calibrated to the global *sensitivity* of the function [8]. However, the injection of noise into an algorithm for preserving differential privacy generally results in a loss of algorithm's accuracy. Since differential privacy is preserved during any post-processing of released output, the accuracy can be increased by denoising the output using statistical estimation theory [9]. The iterative nature of machine learning algorithms poses another challenge, as the iterations causes a high cumulative privacy loss and thus a high amount of noise need to be added to compensate for the privacy loss. To keep track of the privacy loss incurred by successive iterations, the authors in [6] suggest a *moments accountant* method for composition analysis. The moments accountant method is based on the properties of a *privacy loss* random variable. As the method provides a tight bound on the privacy cost of multiple iterations and thus allows for a higher per-iteration privacy budget, it was successfully applied for privacy-preserving variational Bayes [10]. Privacy-preserving distributed deep learning [11, 12] based on distributed stochastic gradient descent offers a solution to preserve each participant's data privacy while still learning from other participants' private data.

Designing of a noise injection mechanism achieving a good trade-off between privacy and accuracy is

obviously of interest and the topic has been studied in the literature [9, 13–18]. Our recent work [19] has suggested a novel entropy based approach for resolving the privacy-utility trade-off for real-valued data matrices. The study in [19] derives mathematically for real-value data matrices the probability density function of noise that minimizes the expected noise magnitude together with satisfying the sufficient conditions for $(\epsilon, \delta)-$differential privacy. We extend the optimal noise adding mechanism of [19] to the datasets consisting of pairs of inputs-outputs vectors. Further, the derived optimal noise adding mechanism is studied with its application for privacy-preserving distributed learning of deep models. The aim is to protect the distributed and deep learning algorithm from an adversary who seeks to gain an information about the training data from learning algorithm's output by perturbing an entry in the training data.

The machine learning methodologies rely on the data for the training of models. The data features, either engineered or extracted using existing trained deep models, might be affected by the noise inherently present in raw data and thus might be imprecise. Further, the mappings between data features and the target variables requires a robustness against uncertainties arising from not only data noise but also other factors such as presence of outliers and choice of non-optimal model structure. The fuzzy sets and rule-based systems are considered as suitable tools for computing with imprecise quantities under the environment of uncertainties. Therefore, the motivation of applying fuzzy theory in deep learning comes from its capability of handling uncertainties in a rigorous mathematical manner. A recent emergence of studies combining fuzzy theory with deep neural networks is observed [20–24, 24–26]. However, the application of fuzzy theory in deep learning remains limited only under the realm of deep neural networks. The fuzzy clustering has been previously [27] considered in private setting, however, the topic has still not attracted the attention of fuzzy machine learning researchers.

Remark 1 (Research Gap) *Though privacy preserving machine learning is currently a hot topic, the fuzzy research community has remained till-now indifferent towards differential privacy formalism. The state-of-art lacks the study of differentially private fuzzy learning algorithms.*

We introduce a privacy-preserving framework for distributed deep fuzzy learning. Specifically, sufficient conditions for $(\epsilon, \delta)-$ differential privacy of the learning algorithm are derived. Following the entropy based approach of [19], the optimal noise distribution that minimizes the expected noise magnitude together with satisfying the sufficient conditions for $(\epsilon, \delta)-$differential privacy is derived. A comparison of the derived noise adding mechanism with the classical Gaussian mechanism is made and a multi-fold reduction (e.g. by more than 4 times in the high privacy regime) in the magnitude of noise over the Gaussian mechanism is observed. To study distributed deep learning problem in private setting, a deep model, formed by a composition of a finite number of Takagi-Sugeno fuzzy filters, is considered. Variational Bayes, a widely used Bayesian inference method, is applied for the learning of deep fuzzy model. We consider a particular configuration of the deep fuzzy model, referred to as *deep fuzzy autoencoder*, for data representation learning. The flexibility and robustness features offered by fuzzy sets and fuzzy rules are next leveraged to facilitate a distributed learning from the data locally owned by different participant. A fuzzy set in the multi-dimensional real space is associated to each local deep fuzzy model. The post-processing invariance property of differential privacy allows to build a global fuzzy rule-based classifier that *aggregates* the fuzzy sets associated to local private deep models using a logical operator.

Remark 2 (Contributions) *This study provides an $(\epsilon, \delta)-$differentially private noise adding mechanism that results in multi-fold reduction in noise magnitude over the classical Gaussian mechanism and thus leads to an increased utility for a given level of privacy. We present an architecture for distributed form of differentially private learning where a privacy wall separates the private local training data from the globally shared data,*

and fuzzy sets and fuzzy rules are used to aggregate robustly the local deep models for building the global model.

2. Distributed Differential Private Deep Learning Models (T3.1)

2.1 An Optimal $(\epsilon, \delta)-$Differentially Private Noise for Real-Valued Datasets

Consider a dataset consisting of N pairs of input and output samples. Let $\mathrm{X} \in \Omega_{\mathrm{x}} \subseteq \mathbb{R}^{n \times N}$ be the $n \times N$ sized matrix representing N number of input samples with each sample having n number of attributes. Similarly, let $\mathrm{Y} \in \Omega_{\mathrm{y}} \subseteq \mathbb{R}^{p \times N}$ be the $p \times N$ sized matrix representing N number of $p-$variate output samples. The dataset can be defined as

$$\mathbf{D} = (\mathrm{X}, \mathrm{Y}). \tag{2.1}$$

A machine learning algorithm uses a dataset for the training of a model. A given machine learning algorithm, training a model using input output samples (X, Y), can be represented by a mapping,

$$\mathcal{A} : \Omega_{\mathrm{x}} \times \Omega_{\mathrm{y}} \rightarrow \mathbf{M}, \tag{2.2}$$

where $\mathbf{M}$ is the model space. For a given dataset $\mathbf{D}$, the machine learning algorithm builds a model $\mathcal{M} \in \mathbf{M}$ such that

$$\begin{aligned} \mathcal{M} &= \mathcal{A}(\mathbf{D}) && (2.3)\\ &= \mathcal{A}(\mathrm{X}, \mathrm{Y}). && (2.4) \end{aligned}$$

The privacy of individual entries in the dataset $\mathbf{D}$ can be preserved via adding a suitable random noise to individual entries before the application of algorithm $\mathcal{A}$ on the dataset. The adding of random noise leads to the private version of algorithm $\mathcal{A}$ as defined in Definition 1.

Definition 1 (Private Algorithm on a Dataset) *Let $\mathcal{A}^{+} : \Omega_{\mathrm{x}} \times \Omega_{\mathrm{y}} \rightarrow Range(\mathcal{A}^{+})$ be a mapping defined as*

$$\mathcal{A}^{+}(\mathrm{X}, \mathrm{Y}) = \mathcal{A}(\mathrm{X} + \mathrm{U}, \mathrm{Y} + \mathrm{V}),\ \mathrm{U} \in \mathbb{R}^{n \times N},\ \mathrm{V} \in \mathbb{R}^{p \times N} \tag{2.5}$$

where U and V are two random noise matrices; and $\mathcal{A} : \Omega_{\mathrm{x}} \times \Omega_{\mathrm{y}} \rightarrow \mathbf{M}$ (where $\mathbf{M}$ is the model space) is a given mapping representing a machine learning algorithm. The range of $\mathcal{A}^{+}$ is as

$$Range(\mathcal{A}^{+}) = \left\{ \mathcal{A}(\mathrm{X} + \mathrm{U}, \mathrm{Y} + \mathrm{V}) \mid \mathrm{X} \in \Omega_{\mathrm{x}}, \mathrm{Y} \in \Omega_{\mathrm{y}}, \mathrm{U} \in \mathbb{R}^{n \times N}, \mathrm{V} \in \mathbb{R}^{p \times N} \right\}.$$

Let u_k^i *be the* $(k,i)-$*th element in* U *and* $f_{\mathrm{u}_k^i}(u)$ *be a probability density function on* u_k^i *with*

$$\int_{\mathbb{R}} f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u = 1. \tag{2.6}$$

It is assumed that u_k^i *and* $\mathrm{u}_k^{i'}$ *are independent from each other for* $i \neq i'$*. Let* v_j^i *be the* $(j,i)-$*th element in* V *and* $f_{\mathrm{v}_j^i}(v)$ *be a probability density function on* v_j^i *with*

$$\int_{\mathbb{R}} f_{\mathrm{v}_j^i}(v)\ \mathrm{d}v = 1. \tag{2.7}$$

It is assumed that v_j^i *and* $\mathrm{v}_j^{i'}$ *are independent from each other for* $i \neq i'$*.*

Algorithm $\mathcal{A}^+$ is intended to be protected from an adversary who seeks to gain an information about the data from algorithm's output via perturbing a single entry in the dataset $\mathbf{D}$. We seek to attain differential privacy for algorithm $\mathcal{A}^+$ against the perturbation in an element of either X or Y such that magnitude of the perturbation is upper bounded by a scalar d. The $d-$adjacency definition for two real-valued datasets is provided in Definition 2.

Definition 2 ($d-$**Adjacency for Real-Valued Datasets)** *Two datasets* $\mathbf{D} = \{(\mathrm{X}, \mathrm{Y})\}$ *(with* $\mathrm{X} \in \Omega_{\mathrm{x}}, \mathrm{Y} \in \Omega_{\mathrm{y}}$*) and* $\mathbf{D}' = \{(\mathrm{X}', \mathrm{Y}')\}$ *(with* $\mathrm{X}' \in \Omega_{\mathrm{x}}, \mathrm{Y}' \in \Omega_{\mathrm{y}}$*) are* $d-$*adjacent if for a given* $d \in \mathbb{R}_+$*, there exists a binary variable* $z \in \{0,1\}$ *such that following holds.*

1. *If* $z = 0$*, then* $\mathrm{Y} = \mathrm{Y}'$ *and every element of* X *is equal to that of* X' *except one element, say* $(k_0, i_0)-$*th element. That is, there exist* $i_0 \in \{1, 2, \cdots, N\}$ *and* $k_0 \in \{1, 2, \cdots, n\}$ *with*

$$\left|\mathrm{x}_k^i - \mathrm{x}_k'^i\right| \leq \begin{cases} d, & \text{if } i = i_0, k = k_0 \\ 0, & \text{otherwise} \end{cases},\ i \in \{1, 2, \cdots, N\},\ k \in \{1, 2, \cdots, n\}$$

   where x_k^i *and* $\mathrm{x}_k'^i$ *denote the* $(k,i)-$*th element of* X *and* X' *respectively.*

2. *If* $z = 1$*, then* $\mathrm{X} = \mathrm{X}'$ *and every element of* Y *is equal to that of* Y' *except one element, say* $(j_0, i_0)-$*th element. That is, there exist* $i_0 \in \{1, 2, \cdots, N\}$ *and* $j_0 \in \{1, 2, \cdots, p\}$ *with*

$$\left|\mathrm{y}_j^i - \mathrm{y}_j'^i\right| \leq \begin{cases} d, & \text{if } i = i_0, j = j_0 \\ 0, & \text{otherwise} \end{cases},\ i \in \{1, 2, \cdots, N\},\ j \in \{1, 2, \cdots, p\}$$

   where y_j^i *and* $\mathrm{y}_j'^i$ *denote the* $(j,i)-$*th element of* Y *and* Y' *respectively.*

Thus, $d-$*adjacency implies that* $\mathbf{D}$ *and* $\mathbf{D}'$ *differ by only one element and the magnitude of the difference is upper bounded by* d*.*

Definition 3 ($(\epsilon, \delta)-$**Differential Privacy for** $\mathcal{A}^+$**)** *The algorithm* $\mathcal{A}^+(\mathbf{D}) = \mathcal{A}^+(\mathrm{X}, \mathrm{Y})$*, defined as per Definition 1, is* $(\epsilon, \delta)-$*differentially private if*

$$Pr\{\mathcal{A}^+(\mathbf{D}) \in \mathcal{O}\} \leq \exp(\epsilon) Pr\{\mathcal{A}^+(\mathbf{D}')) \in \mathcal{O}\} + \delta \tag{2.8}$$

for any measurable set $\mathcal{O} \subseteq Range(\mathcal{A}^+)$ *and for any* $d-$*adjacent datasets pair* $(\mathbf{D}, \mathbf{D}')$*.*

2.1.1 Sufficient Conditions for Differential Privacy

Our first goal is to derive sufficient conditions on $f_{\mathrm{u}_k^i}(u)$ and $f_{\mathrm{v}_j^i}(v)$ for $(\epsilon, \delta)-$differential privacy of $\mathcal{A}^+(\mathbf{D})$. The study of [19] provides sufficient conditions for $(\epsilon, \delta)-$differential privacy of a given machine learning algorithm in the case of matrix-valued datasets. In our problem, the dataset (2.1) is in the form of a pair of real matrices. In this case, the sufficient conditions are provided by Result 1.

Result 1 (Sufficient Conditions for $(\epsilon, \delta)-$Differential Privacy of $\mathcal{A}^+(\mathbf{D})$) *The following conditions on the probability density functions of noises $\mathrm{u}_k^i \in \mathbb{R}$ and $\mathrm{v}_j^i \in \mathbb{R}$ are sufficient to attain $(\epsilon, \delta)-$differential privacy by algorithm $\mathcal{A}^+$:*

$$\int_{\Theta_{\mathrm{u}_k^i}} f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u \geq 1 - \delta \tag{2.9}$$

$$\int_{\Theta_{\mathrm{v}_j^i}} f_{\mathrm{v}_j^i}(v)\ \mathrm{d}v \geq 1 - \delta \tag{2.10}$$

where $\Theta_{\mathrm{u}_k^i} \subseteq \mathbb{R}$ and $\Theta_{\mathrm{v}_j^i} \subseteq \mathbb{R}$ are defined as

$$\Theta_{\mathrm{u}_k^i} \stackrel{\text{def}}{=} \left\{ u \mid \sup_{\hat{d} \in [-d,d]} \frac{f_{\mathrm{u}_k^i - \hat{d}}(u)}{f_{\mathrm{u}_k^i}(u)} \leq \exp(\epsilon),\ f_{\mathrm{u}_k^i}(u) \neq 0,\ \mathrm{u}_k^i \in \mathbb{R} \right\} \tag{2.11}$$

$$\Theta_{\mathrm{v}_j^i} \stackrel{\text{def}}{=} \left\{ v \mid \sup_{\hat{d} \in [-d,d]} \frac{f_{\mathrm{v}_j^i - \hat{d}}(v)}{f_{\mathrm{v}_j^i}(v)} \leq \exp(\epsilon),\ f_{\mathrm{v}_j^i}(v) \neq 0,\ \mathrm{v}_j^i \in \mathbb{R} \right\}. \tag{2.12}$$

Proof: Let $(\mathbf{D}, \mathbf{D}')$ be a pair of $d-$adjacent datasets. As per Definition 2, there exists a binary variable z taking value equal to either 0 or 1 depending upon perturbation has been made in $\mathbf{X}$ or $\mathbf{Y}$. We consider the both cases separately:

The Case of $z = 0$
In this case, there exists a $\hat{d} \in [-d, d]$ for some (i_0, k_0) such that

$$\mathrm{x}_{k_0}^{i_0} = \mathrm{x}_{k_0}'^{i_0} - \hat{d} \tag{2.13}$$

$$\mathrm{x}_k^i = \mathrm{x}_k'^i,\ (i,k) \in \{1, \cdots, N\} \times \{1, \cdots, n\}, (i,k) \neq (i_0, k_0). \tag{2.14}$$

For a given (Y, V), define a set $\mathbf{R}$ as

$$\mathbf{R}(\mathrm{Y}, \mathrm{V}) \stackrel{\text{def}}{=} \{\mathrm{X} + \mathrm{U} \mid \mathcal{A}(\mathrm{X} + \mathrm{U}, \mathrm{Y} + \mathrm{V}) \in \mathcal{O}\}. \tag{2.15}$$

Let $\mathbf{R}_k^i$ be the set of $(k,i)-$th elements of all members in $\mathbf{R}$, i.e.,

$$\mathbf{R}_k^i(\mathrm{Y}, \mathrm{V}) = \left\{ \mathrm{x}_k^i + \mathrm{u}_k^i \mid \mathrm{X} + \mathrm{U} \in \mathbf{R}(\mathrm{Y}, \mathrm{V}) \right\}. \tag{2.16}$$

We have

$$Pr\{\mathcal{A}^+(\mathbf{D}) \in \mathcal{O}\} = Pr\{\mathcal{A}(\mathrm{X} + \mathrm{U}, \mathrm{Y} + \mathrm{V}) \in \mathcal{O}\} \tag{2.17}$$

$$= Pr\{\mathrm{X} + \mathrm{U} \in \mathbf{R}(\mathrm{Y}, \mathrm{V})\} \tag{2.18}$$

$$= \prod_{k=1}^{n} \prod_{i=1}^{N} Pr\{\mathrm{x}_k^i + \mathrm{u}_k^i \in \mathbf{R}_k^i(\mathrm{Y}, \mathrm{V})\}. \tag{2.19}$$

The assumption of independence of elements of U has been used to arrive at (2.19) from (2.18). Equality (2.19) can be expressed as

$$Pr\{\mathcal{A}^{+}(\mathbf{D}) \in \mathcal{O}\} = Pr\{\mathrm{x}_{k_0}^{i_0} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\} \prod_{k,i,k\neq k_0,i\neq i_0} Pr\{\mathrm{x}_k^i + \mathrm{u}_k^i \in \mathbf{R}_k^i(\mathrm{Y},\mathrm{V})\}. \quad (2.20)$$

Using (2.14) in (2.20), we have

$$Pr\{\mathcal{A}^{+}(\mathbf{D}) \in \mathcal{O}\} = Pr\{\mathrm{x}_{k_0}^{i_0} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\} \prod_{k,i,k\neq k_0,i\neq i_0} Pr\{\mathrm{x}_k'^i + \mathrm{u}_k^i \in \mathbf{R}_k^i(\mathrm{Y},\mathrm{V})\}. \quad (2.21)$$

Now, consider

$$\begin{aligned} &Pr\{\mathrm{x}_{k_0}^{i_0} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\} \\ &= Pr\{\mathrm{x}_{k_0}^{i_0} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V}) \mid \mathrm{u}_{k_0}^{i_0} \in (\mathbb{R} \setminus \Theta_{\mathrm{u}_{k_0}^{i_0}})\} + Pr\{\mathrm{x}_{k_0}^{i_0} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V}) \mid \mathrm{u}_{k_0}^{i_0} \in \Theta_{\mathrm{u}_{k_0}^{i_0}}\} \quad (2.22) \end{aligned}$$

Using (2.13) in (2.22), we have

$$\begin{aligned} &Pr\{\mathrm{x}_{k_0}^{i_0} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\} \\ &= Pr\{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V}) \mid \mathrm{u}_{k_0}^{i_0} \in (\mathbb{R} \setminus \Theta_{\mathrm{u}_{k_0}^{i_0}})\} + Pr\{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0} \in \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V}) \mid \mathrm{u}_{k_0}^{i_0} \in \Theta_{\mathrm{u}_{k_0}^{i_0}}\} \quad (2.23) \\ &= \int_{\{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0} \mid \mathrm{u}_{k_0}^{i_0} \in (\mathbb{R}\setminus\Theta_{\mathrm{u}_{k_0}^{i_0}})\} \cap \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})} f_{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u \\ &\quad + \int_{\{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0} \mid \mathrm{u}_{k_0}^{i_0} \in \Theta_{\mathrm{u}_{k_0}^{i_0}}\} \cap \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})} f_{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u. \quad (2.24) \end{aligned}$$

The upper bounds on both terms of the right hand side of (2.24) are derived. First consider,

$$\begin{aligned} &\int_{\{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0} \mid \mathrm{u}_{k_0}^{i_0} \in (\mathbb{R}\setminus\Theta_{\mathrm{u}_{k_0}^{i_0}})\} \cap \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})} f_{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u \\ &\leq \int_{\{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0} \mid \mathrm{u}_{k_0}^{i_0} \in (\mathbb{R}\setminus\Theta_{\mathrm{u}_{k_0}^{i_0}})\}} f_{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u \quad (2.25) \\ &= \int_{\mathbb{R}\setminus\Theta_{\mathrm{u}_{k_0}^{i_0}}} f_{\mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u \quad (2.26) \\ &= \int_{\mathbb{R}} f_{\mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u - \int_{\Theta_{\mathrm{u}_{k_0}^{i_0}}} f_{\mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u. \quad (2.27) \end{aligned}$$

Using (2.6) in (2.27), we get

$$\int_{\{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0} \mid \mathrm{u}_{k_0}^{i_0} \in (\mathbb{R}\setminus\Theta_{\mathrm{u}_{k_0}^{i_0}})\} \cap \mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})} f_{\mathrm{x}_{k_0}'^{i_0} - \hat{d} + \mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u \leq 1 - \int_{\Theta_{\mathrm{u}_{k_0}^{i_0}}} f_{\mathrm{u}_{k_0}^{i_0}}(u)\, \mathrm{d}u. \quad (2.28)$$

It follows from the definition of $\Theta_{\mathrm{u}_{k_0}^{i_0}}$, i.e. from (2.11), that

$$\begin{aligned}
&\int_{\{\mathrm{x}_{k_0}'^{i_0}-\hat{d}+\mathrm{u}_{k_0}^{i_0} \mid \mathrm{u}_{k_0}^{i_0}\in\Theta_{\mathrm{u}_{k_0}^{i_0}}\}\cap\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})} f_{\mathrm{x}_{k_0}'^{i_0}-\hat{d}+\mathrm{u}_{k_0}^{i_0}}(u)\ \mathrm{d}u \\
&\leq \exp(\epsilon)\int_{\{\mathrm{x}_{k_0}'^{i_0}-\hat{d}+\mathrm{u}_{k_0}^{i_0} \mid \mathrm{u}_{k_0}^{i_0}\in\Theta_{\mathrm{u}_{k_0}^{i_0}}\}\cap\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})} f_{\mathrm{x}_{k_0}'^{i_0}+\mathrm{u}_{k_0}^{i_0}}(u)\ \mathrm{d}u \qquad (2.29)\\
&\leq \exp(\epsilon)\int_{\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})} f_{\mathrm{x}_{k_0}'^{i_0}+\mathrm{u}_{k_0}^{i_0}}(u)\ \mathrm{d}u \qquad (2.30)\\
&= \exp(\epsilon)Pr\{\mathrm{x}_{k_0}'^{i_0}+\mathrm{u}_{k_0}^{i_0}\in\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\}. \qquad (2.31)
\end{aligned}$$

Using (2.28) and (2.31) in (2.24), we have

$$Pr\{\mathrm{x}_{k_0}^{i_0}+\mathrm{u}_{k_0}^{i_0}\in\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\}\leq 1-\int_{\Theta_{\mathrm{u}_{k_0}^{i_0}}} f_{\mathrm{u}_{k_0}^{i_0}}(u)\ \mathrm{d}u+\exp(\epsilon)Pr\{\mathrm{x}_{k_0}'^{i_0}+\mathrm{u}_{k_0}^{i_0}\in\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\}. \qquad (2.32)$$

Under condition (2.9), inequality (2.32) leads to

$$Pr\{\mathrm{x}_{k_0}^{i_0}+\mathrm{u}_{k_0}^{i_0}\in\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\}\leq\delta+\exp(\epsilon)Pr\{\mathrm{x}_{k_0}'^{i_0}+\mathrm{u}_{k_0}^{i_0}\in\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\}. \qquad (2.33)$$

That is,

$$\begin{aligned}
&Pr\{\mathrm{x}_{k_0}^{i_0}+\mathrm{u}_{k_0}^{i_0}\in\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\}\prod_{k,i,k\neq k_0,i\neq i_0}Pr\{\mathrm{x}_k'^{i}+\mathrm{u}_k^{i}\in\mathbf{R}_k^{i}(\mathrm{Y},\mathrm{V})\}\\
&\leq\delta\prod_{k,i,k\neq k_0,i\neq i_0}Pr\{\mathrm{x}_k'^{i}+\mathrm{u}_k^{i}\in\mathbf{R}_k^{i}(\mathrm{Y},\mathrm{V})\}\\
&\quad+\exp(\epsilon)Pr\{\mathrm{x}_{k_0}'^{i_0}+\mathrm{u}_{k_0}^{i_0}\in\mathbf{R}_{k_0}^{i_0}(\mathrm{Y},\mathrm{V})\}\prod_{k,i,k\neq k_0,i\neq i_0}Pr\{\mathrm{x}_k'^{i}+\mathrm{u}_k^{i}\in\mathbf{R}_k^{i}(\mathrm{Y},\mathrm{V})\} \qquad (2.34)\\
&=\delta\prod_{k,i,k\neq k_0,i\neq i_0}Pr\{\mathrm{x}_k'^{i}+\mathrm{u}_k^{i}\in\mathbf{R}_k^{i}(\mathrm{Y},\mathrm{V})\}+\exp(\epsilon)\prod_{k=1}^{n}\prod_{i=1}^{N}Pr\{\mathrm{x}_k'^{i}+\mathrm{u}_k^{i}\in\mathbf{R}_k^{i}(\mathrm{Y},\mathrm{V})\} \qquad (2.35)\\
&\leq\delta+\exp(\epsilon)\prod_{k=1}^{n}\prod_{i=1}^{N}Pr\{\mathrm{x}_k'^{i}+\mathrm{u}_k^{i}\in\mathbf{R}_k^{i}(\mathrm{Y},\mathrm{V})\}. \qquad (2.36)
\end{aligned}$$

Using inequality (2.36) in (2.21), we have

$$\begin{aligned}
Pr\{\mathcal{A}^{+}(\mathbf{D})\in\mathcal{O}\}&\leq\delta+\exp(\epsilon)\prod_{k=1}^{n}\prod_{i=1}^{N}Pr\{\mathrm{x}_k'^{i}+\mathrm{u}_k^{i}\in\mathbf{R}_k^{i}(\mathrm{Y},\mathrm{V})\} \qquad (2.37)\\
&=\delta+\exp(\epsilon)Pr\{\mathrm{X}'+\mathrm{U}\in\mathbf{R}(\mathrm{Y},\mathrm{V})\}. \qquad (2.38)
\end{aligned}$$

In the case of $z=0$, $\mathrm{Y}=\mathrm{Y}'$ and thus we replace Y by Y' in (2.38) to get

$$\begin{aligned}
Pr\{\mathcal{A}^{+}(\mathbf{D})\in\mathcal{O}\}&\leq\delta+\exp(\epsilon)Pr\{\mathrm{X}'+\mathrm{U}\in\mathbf{R}(\mathrm{Y}',\mathrm{V})\} \qquad (2.39)\\
&=\delta+\exp(\epsilon)Pr\{\mathcal{A}\left(\mathrm{X}'+\mathrm{U},\mathrm{Y}'+\mathrm{V}\right)\in\mathcal{O}\} \qquad (2.40)\\
&=\delta+\exp(\epsilon)Pr\{\mathcal{A}^{+}\left(\mathbf{D}'\right)\in\mathcal{O}\}. \qquad (2.41)
\end{aligned}$$

For $z=0$, (2.8) is satisfied and thus $\mathcal{A}^{+}(\mathbf{D})$ is $(\epsilon,\delta)-$differentially private.

The Case of $z = 1$
In this case, there exists a $\hat{d} \in [-d, d]$ for some (i_0, j_0) such that

$$\mathrm{y}_{j_0}^{i_0} = \mathrm{y}_{j_0}'^{i_0} - \hat{d} \tag{2.42}$$
$$\mathrm{y}_j^i = \mathrm{y}_j'^i,\ (i,j) \in \{1, \cdots, N\} \times \{1, \cdots, p\}, (i,j) \neq (i_0, j_0). \tag{2.43}$$

For a given (X, U), define a set $\mathbf{S}$ as

$$\mathbf{S}(\mathrm{X}, \mathrm{U}) \stackrel{\text{def}}{=} \{\mathrm{Y} + \mathrm{V} \mid \mathcal{A}(\mathrm{X} + \mathrm{U}, \mathrm{Y} + \mathrm{V}) \in \mathcal{O}\}. \tag{2.44}$$

Let $\mathbf{S}_j^i$ as the set of $(j,i)-$th elements of all members in $\mathbf{S}$, i.e.,

$$\mathbf{S}_j^i(\mathrm{X}, \mathrm{U}) \stackrel{\text{def}}{=} \left\{\mathrm{y}_j^i + \mathrm{v}_j^i \mid \mathrm{Y} + \mathrm{V} \in \mathbf{S}(\mathrm{X}, \mathrm{U})\right\}. \tag{2.45}$$

We have

$$Pr\{\mathcal{A}^+(\mathbf{D}) \in \mathcal{O}\} = Pr\{\mathcal{A}(\mathrm{X} + \mathrm{U}, \mathrm{Y} + \mathrm{V}) \in \mathcal{O}\} \tag{2.46}$$
$$= Pr\{\mathrm{Y} + \mathrm{V} \in \mathbf{S}(\mathrm{X}, \mathrm{U})\} \tag{2.47}$$
$$= \prod_{j=1}^{p} \prod_{i=1}^{N} Pr\{\mathrm{y}_j^i + \mathrm{v}_j^i \in \mathbf{S}_j^i(\mathrm{X}, \mathrm{U})\}. \tag{2.48}$$

The assumption of independence of elements of V has been used to arrive at (2.48) from (2.47). Equality (2.48) can be expressed as

$$Pr\{\mathcal{A}^+(\mathbf{D}) \in \mathcal{O}\} = Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U})\} \prod_{j,i,j\neq j_0, i\neq i_0} Pr\{\mathrm{y}_j^i + \mathrm{v}_j^i \in \mathbf{S}_j^i(\mathrm{X}, \mathrm{U})\}. \tag{2.49}$$

Using (2.43) in (2.49), we have

$$Pr\{\mathcal{A}^+(\mathbf{D}) \in \mathcal{O}\} = Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U})\} \prod_{j,i,j\neq j_0, i\neq i_0} Pr\{\mathrm{y}_j'^i + \mathrm{v}_j^i \in \mathbf{S}_j^i(\mathrm{X}, \mathrm{U})\}. \tag{2.50}$$

Now, consider

$$Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U})\}$$
$$= Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U}) \mid \mathrm{v}_{j_0}^{i_0} \in (\mathbb{R} \setminus \Theta_{\mathrm{v}_{j_0}^{i_0}})\} + Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U}) \mid \mathrm{v}_{j_0}^{i_0} \in \Theta_{\mathrm{v}_{j_0}^{i_0}}\}. \tag{2.51}$$

Using (2.42) in (2.51), we have

$$Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U})\}$$
$$= Pr\{\mathrm{y}_{j_0}'^{i_0} - \hat{d} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U}) \mid \mathrm{v}_{j_0}^{i_0} \in (\mathbb{R} \setminus \Theta_{\mathrm{v}_{j_0}^{i_0}})\} + Pr\{\mathrm{y}_{j_0}'^{i_0} - \hat{d} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U}) \mid \mathrm{v}_{j_0}^{i_0} \in \Theta_{\mathrm{v}_{j_0}^{i_0}}\} \tag{2.52}$$
$$= \int_{\{\mathrm{y}_{j_0}'^{i_0} - \hat{d} + \mathrm{v}_{j_0}^{i_0} \mid \mathrm{v}_{j_0}^{i_0} \in (\mathbb{R} \setminus \Theta_{\mathrm{v}_{j_0}^{i_0}})\} \cap \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U})} f_{\mathrm{y}_{j_0}'^{i_0} - \hat{d} + \mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v$$
$$+ \int_{\{\mathrm{y}_{j_0}'^{i_0} - \hat{d} + \mathrm{v}_{j_0}^{i_0} \mid \mathrm{v}_{j_0}^{i_0} \in \Theta_{\mathrm{v}_{j_0}^{i_0}}\} \cap \mathbf{S}_{j_0}^{i_0}(\mathrm{X}, \mathrm{U})} f_{\mathrm{y}_{j_0}'^{i_0} - \hat{d} + \mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v. \tag{2.53}$$

The upper bounds on both terms of the right hand side of (2.53) are derived. First consider,

$$\int_{\{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0} \mid \mathrm{v}_{j_0}^{i_0}\in(\mathbb{R}\setminus\Theta_{\mathrm{v}_{j_0}^{i_0}})\}\cap\mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})} f_{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v$$

$$\leq \int_{\{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0} \mid \mathrm{v}_{j_0}^{i_0}\in(\mathbb{R}\setminus\Theta_{\mathrm{v}_{j_0}^{i_0}})\}} f_{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v \tag{2.54}$$

$$= \int_{\mathbb{R}\setminus\Theta_{\mathrm{v}_{j_0}^{i_0}}} f_{\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v \tag{2.55}$$

$$= \int_{\mathbb{R}} f_{\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v - \int_{\Theta_{\mathrm{v}_{j_0}^{i_0}}} f_{\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v. \tag{2.56}$$

Using (2.7) in (2.56), we get

$$\int_{\{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0} \mid \mathrm{v}_{j_0}^{i_0}\in(\mathbb{R}\setminus\Theta_{\mathrm{v}_{j_0}^{i_0}})\}\cap\mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})} f_{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v \leq 1 - \int_{\Theta_{\mathrm{v}_{j_0}^{i_0}}} f_{\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v. \tag{2.57}$$

It follows from the definition of $\Theta_{\mathrm{v}_{j_0}^{i_0}}$, i.e. from (2.12), that

$$\int_{\{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0} \mid \mathrm{v}_{j_0}^{i_0}\in\Theta_{\mathrm{v}_{j_0}^{i_0}}\}\cap\mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})} f_{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v$$

$$\leq \exp(\epsilon) \int_{\{\mathrm{y}_{j_0}^{\prime i_0}-\hat{d}+\mathrm{v}_{j_0}^{i_0} \mid \mathrm{v}_{j_0}^{i_0}\in\Theta_{\mathrm{v}_{j_0}^{i_0}}\}\cap\mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})} f_{\mathrm{y}_{j_0}^{\prime i_0}+\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v \tag{2.58}$$

$$\leq \exp(\epsilon) \int_{\mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})} f_{\mathrm{y}_{j_0}^{\prime i_0}+\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v \tag{2.59}$$

$$= \exp(\epsilon) Pr\{\mathrm{y}_{j_0}^{\prime i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})\}. \tag{2.60}$$

Using (2.57) and (2.60) in (2.53), we have

$$Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})\} \leq 1 - \int_{\Theta_{\mathrm{v}_{j_0}^{i_0}}} f_{\mathrm{v}_{j_0}^{i_0}}(v)\ \mathrm{d}v + \exp(\epsilon) Pr\{\mathrm{y}_{j_0}^{\prime i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})\}. \tag{2.61}$$

Under condition (2.10), inequality (2.61) leads to

$$Pr\{\mathrm{y}_{j_0}^{i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})\} \leq \delta + \exp(\epsilon) Pr\{\mathrm{y}_{j_0}^{\prime i_0} + \mathrm{v}_{j_0}^{i_0} \in \mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})\}. \tag{2.62}$$

That is,

$$
\begin{aligned}
&Pr\{\mathrm{y}_{j_0}^{i_0}+\mathrm{v}_{j_0}^{i_0}\in\mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})\}\prod_{j,i,j\neq j_0,i\neq i_0}Pr\{\mathrm{y}_j'^i+\mathrm{v}_j^i\in\mathbf{S}_j^i(\mathrm{X},\mathrm{U})\}\\
&\leq\delta\prod_{j,i,j\neq j_0,i\neq i_0}Pr\{\mathrm{y}_j'^i+\mathrm{v}_j^i\in\mathbf{S}_j^i(\mathrm{X},\mathrm{U})\}\\
&\quad+\exp(\epsilon)Pr\{\mathrm{y}_{j_0}'^{i_0}+\mathrm{v}_{j_0}^{i_0}\in\mathbf{S}_{j_0}^{i_0}(\mathrm{X},\mathrm{U})\}\prod_{j,i,j\neq j_0,i\neq i_0}Pr\{\mathrm{y}_j'^i+\mathrm{v}_j^i\in\mathbf{S}_j^i(\mathrm{X},\mathrm{U})\} &(2.63)\\
&=\delta\prod_{j,i,j\neq j_0,i\neq i_0}Pr\{\mathrm{y}_j'^i+\mathrm{v}_j^i\in\mathbf{S}_j^i(\mathrm{X},\mathrm{U})\}+\exp(\epsilon)\prod_{j=1}^{p}\prod_{i=1}^{N}Pr\{\mathrm{y}_j'^i+\mathrm{v}_j^i\in\mathbf{S}_j^i(\mathrm{X},\mathrm{U})\} &(2.64)\\
&\leq\delta+\exp(\epsilon)\prod_{j=1}^{p}\prod_{i=1}^{N}Pr\{\mathrm{y}_j'^i+\mathrm{v}_j^i\in\mathbf{S}_j^i(\mathrm{X},\mathrm{U})\}. &(2.65)
\end{aligned}
$$

Using inequality (2.65) in (2.50), we have

$$
\begin{aligned}
Pr\{\mathcal{A}^+(\mathbf{D})\in\mathcal{O}\}&\leq\delta+\exp(\epsilon)\prod_{j=1}^{p}\prod_{i=1}^{N}Pr\{\mathrm{y}_j'^i+\mathrm{v}_j^i\in\mathbf{S}_j^i(\mathrm{X},\mathrm{U})\} &(2.66)\\
&=\delta+\exp(\epsilon)Pr\{\mathrm{Y}'+\mathrm{V}\in\mathbf{S}(\mathrm{X},\mathrm{U})\}. &(2.67)
\end{aligned}
$$

In the case of $z=1$, $\mathrm{X}=\mathrm{X}'$ and thus we replace X by X' in (2.67) to get

$$
\begin{aligned}
Pr\{\mathcal{A}^+(\mathbf{D})\in\mathcal{O}\}&\leq\delta+\exp(\epsilon)Pr\{\mathrm{Y}'+\mathrm{V}\in\mathbf{S}(\mathrm{X}',\mathrm{U})\} &(2.68)\\
&=\delta+\exp(\epsilon)Pr\{\mathcal{A}\left(\mathrm{X}'+\mathrm{U},\mathrm{Y}'+\mathrm{V}\right)\in\mathcal{O}\} &(2.69)\\
&=\delta+\exp(\epsilon)Pr\{\mathcal{A}^+\left(\mathbf{D}'\right)\in\mathcal{O}\}. &(2.70)
\end{aligned}
$$

For $z=1$, (2.8) is satisfied and thus $\mathcal{A}^+(\mathbf{D})$ is $(\epsilon,\delta)-$differentially private.

Hence, (2.9), (2.10), (2.11), (2.12) are sufficient conditions for $(\epsilon,\delta)-$differential privacy. ■

Remark 3 (Sufficient Conditions for $\epsilon-$Differential Privacy) *The sufficient conditions for $\epsilon-$differential privacy follow from (2.9), (2.10) taking $\delta=0$ as*

$$\int_{\Theta_{\mathrm{u}_k^i}}f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u=1 \tag{2.71}$$

$$\int_{\Theta_{\mathrm{v}_j^i}}f_{\mathrm{v}_j^i}(v)\ \mathrm{d}v=1 \tag{2.72}$$

where $\Theta_{\mathrm{u}_k^i}$ and $\Theta_{\mathrm{v}_j^i}$ are defined as in (2.11) and (2.12) respectively. The equality in (2.71) and (2.72) is due to the fact that the integral of any probability density function over a subset can't exceed unity. For the case when $\Theta_{\mathrm{u}_k^i}=\mathbb{R}$ and $\Theta_{\mathrm{v}_j^i}=\mathbb{R}$, the sufficient conditions can be expressed as

$$\sup_{\hat{d}\in[-d,d]}\frac{f_{\mathrm{u}_k^i-\hat{d}}(u)}{f_{\mathrm{u}_k^i}(u)}\leq\exp(\epsilon),\ f_{\mathrm{u}_k^i}(u)\neq0,\ \mathrm{u}_k^i\in\mathbb{R} \tag{2.73}$$

$$\sup_{\hat{d}\in[-d,d]}\frac{f_{\mathrm{v}_j^i-\hat{d}}(v)}{f_{\mathrm{v}_j^i}(v)}\leq\exp(\epsilon),\ f_{\mathrm{v}_j^i}(v)\neq0,\ \mathrm{v}_j^i\in\mathbb{R}. \tag{2.74}$$

2.1.2 An Optimal Differentially Private Noise

Having derived the sufficient conditions for differential privacy, our approach to derive optimal differentially private noise consists of following steps.

1. For a given level of entropy, the functional form of density function minimizing the expected noise magnitude is derived. This is presented as Result 2.

2. The entropy level is optimized based on the minimization of expected noise magnitude constrained to $\epsilon-$differential privacy. This is presented as Result 3.

3. The solution space for optimization of $(\epsilon, \delta)-$differentially private noise consists of discontinuous distributions having an arbitrary probability mass at an arbitrary point on the domain of continuous distributions satisfying the sufficient conditions for $\epsilon-$differential privacy. The density function, minimizing the expected noise magnitude in the solution space under $(\epsilon, \delta)-$differential privacy constraints, is derived. This is presented as Result 4.

Result 2 (Minimum Magnitude for a Given Entropy Level) *The probability density functions of noises u_k^i and v_j^i that, for a given level of entropy h, minimizes expected noise magnitudes are given as*

$$f^*_{\mathrm{u}_k^i}(u;h) = \frac{1}{\exp(h-1)}\exp(-\frac{2|u|}{\exp(h-1)}), \tag{2.75}$$

$$f^*_{\mathrm{v}_j^i}(v;h) = \frac{1}{\exp(h-1)}\exp(-\frac{2|v|}{\exp(h-1)}), \tag{2.76}$$

where h is the given entropy level. The expected noise magnitudes are given as

$$E_{f^*_{\mathrm{u}_k^i}}\left[|u|\right](h) = \frac{1}{2}\exp(h-1), \tag{2.77}$$

$$E_{f^*_{\mathrm{v}_j^i}}\left[|v|\right](h) = \frac{1}{2}\exp(h-1). \tag{2.78}$$

Proof: The proof follows from [19]. For the sake of completeness, the proof is provided in 6.1. ■

Result 3 (An Optimal $\epsilon-$Differentially Private Noise) *The probability density functions of noises, minimizing the expected noise magnitudes together with satisfying the sufficient conditions for $\epsilon-$differential privacy, are given as*

$$f^*_{\mathrm{u}_k^i}(u) = \frac{\epsilon}{2d}\exp(-\frac{\epsilon}{d}|u|) \tag{2.79}$$

$$f^*_{\mathrm{v}_j^i}(v) = \frac{\epsilon}{2d}\exp(-\frac{\epsilon}{d}|v|). \tag{2.80}$$

The optimal values of expected noise magnitudes are given as

$$E_{f^*_{\mathrm{u}_k^i}}\left[|u|\right] = \frac{d}{\epsilon} \tag{2.81}$$

$$E_{f^*_{\mathrm{v}_j^i}}\left[|v|\right] = \frac{d}{\epsilon}. \tag{2.82}$$

Proof: The proof is provided in 6.2. The proof follows the steps as in [19]. ■

Result 3 justifies the widely used Laplacian distribution for $\epsilon-$differential privacy.

Result 4 (An Optimal $(\epsilon, \delta)-$Differentially Private Noise) *The probability density functions of noises, minimizing the expected noise magnitude together with satisfying the sufficient conditions for $(\epsilon, \delta)-$differential privacy, are given as*

$$f^*_{\mathrm{u}^i_k}(u) = \begin{cases} \delta Dirac\delta(u), & u = 0 \\ (1-\delta)\frac{\epsilon}{2d}\exp(-\frac{\epsilon}{d}|u|), & u \in \mathbb{R} \setminus \{0\} \end{cases} \tag{2.83}$$

$$f^*_{\mathrm{v}^i_j}(v) = \begin{cases} \delta Dirac\delta(v), & v = 0 \\ (1-\delta)\frac{\epsilon}{2d}\exp(-\frac{\epsilon}{d}|v|), & v \in \mathbb{R} \setminus \{0\} \end{cases} \tag{2.84}$$

where $Dirac\delta(u)$ is Dirac delta function satisfying $\int_{-\infty}^{\infty} Dirac\delta(u)\ \mathrm{d}u = 1$. The optimal values of expected noise magnitudes are given as

$$E_{f^*_{\mathrm{u}^i_k}}[|u|] = (1-\delta)\frac{d}{\epsilon} \tag{2.85}$$

$$E_{f^*_{\mathrm{v}^i_j}}[|v|] = (1-\delta)\frac{d}{\epsilon}. \tag{2.86}$$

Proof: The proof is provided in 6.3. The proof follows the steps as in [19]. ■

Remark 4 (Generating Random Samples from Optimal $(\epsilon, \delta)-$Differentially Private Noise) *The method of "inverse transform sampling" can be used to generate random samples from $f^*_{\mathrm{u}^i_k}(u)$ and $f^*_{\mathrm{v}^i_j}(v)$ using their cumulative distribution functions. The cumulative distribution function of $f^*_{\mathrm{u}^i_k}(u)$ is given as*

$$F_{\mathrm{u}^i_k}(u) = \begin{cases} \frac{1-\delta}{2}\exp(\frac{\epsilon}{d}u), & u < 0 \\ \frac{1+\delta}{2}, & u = 0 \\ 1 - \frac{1-\delta}{2}\exp(-\frac{\epsilon}{d}u), & u > 0 \end{cases} \tag{2.87}$$

The inverse cumulative distribution function is given as

$$F^{-1}_{\mathrm{u}^i_k}(t) = \begin{cases} \frac{d}{\epsilon}\log(\frac{2t}{1-\delta}), & t < \frac{1-\delta}{2} \\ 0, & t \in [\frac{1-\delta}{2}, \frac{1+\delta}{2}] \\ -\frac{d}{\epsilon}\log(\frac{2(1-t)}{1-\delta}), & t > \frac{1+\delta}{2} \end{cases}, \ t \in (0,1). \tag{2.88}$$

Thus, via generating random samples from the uniform distribution on $(0, 1)$ and using (2.88), the noise additive mechanism can be implemented.

2.1.3 Comparison with Classical Gaussian Mechanism

This subsection is dedicated to the comparison of derived $(\epsilon, \delta)-$differentially private noise adding mechanism with that of otherwise widely used Gaussian mechanism. The Gaussian mechanism adds Gaussian distributed noise calibrated to the sensitivity of query function as described in Definition 4 and Result 5.

Definition 4 (Global L_2 Sensitivity of a Vector-Valued Function of a Dataset) *For a vector-valued function of a dataset $f(\mathbf{D})$, the global L_2 sensitivity of f is defined as*

$$\Delta f \stackrel{\text{def}}{=} \max_{\mathbf{D},\mathbf{D}'} \|f(\mathbf{D}) - f(\mathbf{D}')\|_2 \tag{2.89}$$

for all $\mathbf{D}, \mathbf{D}'$ differing on a single entry over the entire dataset domain.

Result 5 (Gaussian Mechanism [4] for $f(\mathbf{D})$) *For any $\epsilon, \delta \in (0,1)$, the mechanism*

$$f_i^+(\mathbf{D}) = f_i(\mathbf{D}) + e_i, \text{ where } e_i \sim \mathcal{N}(0, \sigma_e^2) \tag{2.90}$$

with $\sigma_e \geq \Delta f \sqrt{2\log(1.25/\delta)}/\epsilon$, is $(\epsilon, \delta)-$differentially private. Here, f_i is the $i-$th element of $f(\mathbf{D})$.

Proof: The proof follows from [4]. ■

The dataset $\mathbf{D}$ in our problem consists of a pair of matrices (X, Y) such that $\mathrm{X} \in \Omega_{\mathrm{x}} \subseteq \mathbb{R}^{n \times N}$ and $\mathrm{Y} \in \Omega_{\mathrm{y}} \subseteq \mathbb{R}^{p \times N}$. For an application of Gaussian mechanism on $\mathbf{D}$, define a vector-valued function, $f : \Omega_{\mathrm{x}} \times \Omega_{\mathrm{y}} \to \mathbb{R}^{(n+p)N}$, as

$$f(\mathbf{D}) = f(\mathrm{X}, \mathrm{Y}) \tag{2.91}$$

$$\stackrel{\text{def}}{=} \begin{bmatrix} \mathrm{vec}(\mathrm{X}) \\ \mathrm{vec}(\mathrm{Y}) \end{bmatrix}, \tag{2.92}$$

where $\mathrm{vec}(\cdot)$ is the vectorization operation on a matrix. If $(\mathbf{D}, \mathbf{D}')$ is a $d-$adjacent dataset pair, the global L_2 sensitivity of $f(\mathbf{D})$ is given as

$$\Delta f = \max_{(\mathrm{X},\mathrm{Y}),(\mathrm{X}',\mathrm{Y}')} \left\| \begin{bmatrix} \mathrm{vec}(\mathrm{X}) - \mathrm{vec}(\mathrm{X}') \\ \mathrm{vec}(\mathrm{Y}) - \mathrm{vec}(\mathrm{Y}') \end{bmatrix} \right\|_2 \tag{2.93}$$

$$= d. \tag{2.94}$$

Result 6 (Gaussian Mechanism for $\mathbf{D}$) *For any $\epsilon, \delta \in (0,1)$, the mechanism*

$$\mathrm{x}_k^{+i} = \mathrm{x}_k^i + \mathrm{u}_k^i, \ \mathrm{u}_k^i \sim \mathcal{N}(0, \sigma^2) \tag{2.95}$$

$$\mathrm{y}_j^{+i} = \mathrm{y}_j^i + \mathrm{v}_j^i, \ \mathrm{v}_j^i \sim \mathcal{N}(0, \sigma^2) \tag{2.96}$$

with $\sigma = d\sqrt{2\log(1.25/\delta)}/\epsilon$, is $(\epsilon, \delta)-$differentially private. Here, x_k^i is $(k,i)-$th element of X and y_j^i is $(j,i)-$th element of Y. The expected noise magnitudes are given as

$$E[|\mathrm{u}_k^i|] = \frac{2}{\sqrt{\pi}} \frac{d}{\epsilon} \sqrt{\log(1.25/\delta)} \tag{2.97}$$

$$E[|\mathrm{v}_j^i|] = \frac{2}{\sqrt{\pi}} \frac{d}{\epsilon} \sqrt{\log(1.25/\delta)}. \tag{2.98}$$

Proof: The differential privacy follows directly from Result 5 considering that $\Delta f = d$. The expectation of magnitude of noise u_k^i is given as

$$E[|\mathrm{u}_k^i|] = 2\int_0^\infty u \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{u^2}{2\sigma^2})\ \mathrm{d}u \tag{2.99}$$

$$= \sigma\sqrt{\frac{2}{\pi}} \tag{2.100}$$

$$= \frac{2}{\sqrt{\pi}}\frac{d}{\epsilon}\sqrt{\log\left(1.25/\delta\right)}. \tag{2.101}$$

■

For a comparison with Gaussian mechanism, the ratio of expected noise magnitude of classical Gaussian mechanism to that of proposed mechanism (i.e. Result 4) can be calculated using (2.97) and (2.85) as

$$r(\delta) = \frac{2}{(1-\delta)\sqrt{\pi}}\sqrt{\log\left(1.25/\delta\right)}. \tag{2.102}$$

The ratio $r(\delta)$ is plotted over δ in Fig. 2.1. Fig. 2.1 shows a multi-fold multiplicative gain over the Gaussian

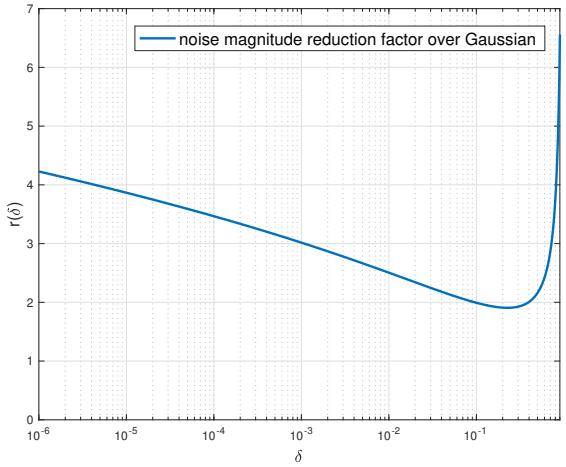


Figure 2.1: Ratio of expected noise magnitude of the classical Gaussian mechanism to that of proposed mechanism.

mechanism for minimizing the noise magnitude. The noise magnitude reduction factor is increasingly more pronounced in the high privacy regime (i.e. low δ), however, also shoots up in the low privacy regime as $\delta \to 1$. It is concluded that the derived optimal $(\epsilon, \delta)-$ differentially private mechanism reduces the noise magnitude by more than 4 times in the high privacy regime over the Gaussian mechanism.

2.2 Variational Bayesian Deep Fuzzy Model

2.2.1 Background and Mathematical Formulation of a Deep Fuzzy Model

Definition 5 (Multivariate Membership Function) *A $n-$dimensional fuzzy set $\mathbf{A}^n$ defined on a universe of discourse $\mathbf{X}^n$ is characterized by a multivariate membership function $\mu_{\mathbf{A}^n} : \mathbf{X}^n \to [0,1]$, i.e.,*

$$\mathbf{A}^n = \{((x_1, \cdots, x_n), \mu_{\mathbf{A}^n}(x_1, \cdots, x_n)) \mid x_k \in \mathbf{X}, k \in \{1, 2, \cdots, n\}\}. \tag{2.103}$$

2.2.1.1 A Review of Zero-Order Takagi-Sugeno Fuzzy Filter

Consider a fuzzy filter ($\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$) that maps $n-$dimensional real-space to $1-$dimensional real-line. We consider a particular form of Takagi-Sugeno filter where M different $n-$dimensional fuzzy sets are defined on the input space, and corresponding to each fuzzy set, there exists a fuzzy rule of the following type

$$m-\text{th rule} \quad : \quad \text{If } x \text{ is } \mathbf{A}_m^n, \text{ then } \mathcal{F}(x) = c_m$$

where $x = [\, x_1 \ \cdots \ x_n \,]^T \in \mathbb{R}^n$, $c_m \in \mathbb{R}$, $m \in \{1, 2, \cdots, M\}$, and the fuzzy set $\mathbf{A}_m^n$ is defined, without loss of generality, with the following Gaussian membership function

$$\mu_{\mathbf{A}_m^n}(x) = \exp\left(-\frac{1}{2}\|x - a^m\|_W^2\right) \tag{2.104}$$

where $a^m \in \mathbb{R}^n$ is the mean of $\mathbf{A}_m^n$, $W \in \mathbb{R}^{n\times n}(W > 0)$, and $\|x\|_P^2 \stackrel{\text{def}}{=} x^T P x$. For a given input $x \in \mathbb{R}^n$, the *degree of fulfillment* of the $m-$th rule is given by $\mu_{\mathbf{A}_m^n}(x)$. The output of the filter to input vector x is computed by taking the weighted average of the output provided by each rule, i.e.,

$$\mathcal{F}(x) = \frac{\sum_{m=1}^M \mu_{\mathbf{A}_m^n}(x) c_m}{\sum_{m=1}^M \mu_{\mathbf{A}_m^n}(x)}. \tag{2.105}$$

We introduce the following notation

$$\mathbf{x} = \left\{x^i \mid x^i \in \mathbb{R}^n, i \in \{1, \cdots, N\}\right\} \tag{2.106}$$

$$\mathbf{a} = \{a^m \mid a^m \in \mathbb{R}^n, m \in \{1, \cdots, M\}\} \tag{2.107}$$

$$\mathrm{f} = \left[\, \mathcal{F}(x^1) \ \cdots \ \mathcal{F}(x^N) \,\right]^T \in \mathbb{R}^N \tag{2.108}$$

$$\alpha = [\, c_1 \ \cdots \ c_M \,]^T \in \mathbb{R}^M. \tag{2.109}$$

Let $K_{\mathbf{xa}} \in \mathbb{R}^{N\times M}$ be a matrix whose $(i,m)-$th element is given as

$$(K_{\mathbf{xa}}(W))_{i,m} = \frac{\exp\left(-\frac{1}{2}\left\|x^i - a^m\right\|_W^2\right)}{\sum_{m=1}^M \exp\left(-\frac{1}{2}\left\|x^i - a^m\right\|_W^2\right)}. \tag{2.110}$$

It follows from (2.105) that

$$\mathrm{f} = K_{\mathbf{xa}}\alpha. \tag{2.111}$$

Expression (2.111) shows that output of the filter is linear in consequent parameters α.

Remark 5 (Membership Function Type) *Our approach is independent of the choice of membership function type while the Gaussian type has been considered as an example. To compare different membership function types is beyond the focus of current study.*

2.2.1.2 A Deep Fuzzy Model

Definition 6 (Fuzzy Model (FM)) *A fuzzy model, $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^p$, maps an input vector $x \in \mathbb{R}^n$ to the output vector $\mathcal{G}(x) \in \mathbb{R}^p$ given as*

$$\mathcal{G}(x) = [\, \mathcal{F}_1(x) \ \cdots \ \mathcal{F}_p(x) \,]^T \in \mathbb{R}^p \tag{2.112}$$

where $\mathcal{F}_j$ ($j \in \{1, 2, \cdots, p\}$) is a Takagi-Sugeno fuzzy filter, with consequent parameters represented by $\alpha_j = [\, c_{j,1} \ \cdots \ c_{j,M} \,]^T \in \mathbb{R}^M$, such that

$$\mathcal{F}_j(x) = \frac{\sum_{m=1}^{M} \mu_{\mathbf{A}_m^n}(x) c_{j,m}}{\sum_{m=1}^{M} \mu_{\mathbf{A}_m^n}(x)}. \tag{2.113}$$

For $j \in \{1, 2, \cdots, p\}$, define

$$\mathrm{f}_j = \left[\, \mathcal{F}_j(x^1) \ \cdots \ \mathcal{F}_j(x^N) \,\right]^T \in \mathbb{R}^N \tag{2.114}$$

to express f_j, similar to (2.111), as

$$\mathrm{f}_j = K_{\mathbf{xa}} \alpha_j. \tag{2.115}$$

Definition 7 (Deep Fuzzy Model (DFM)) *A deep fuzzy model, $\mathcal{D} : \mathbb{R}^n \to \mathbb{R}^p$, maps an input vector $x \in \mathbb{R}^n$ to $\mathcal{D}(x) \in \mathbb{R}^p$ such that*

$$\mathcal{D}(x) = \mathcal{G}_L(\cdots V^T \mathcal{G}_3 \left(V^T \mathcal{G}_2 \left(V^T \mathcal{G}_1(x) \right) \right) \cdots), \tag{2.116}$$

where $\mathcal{G}_l$ ($l \in \{1, \cdots, L\}$) is a FM (Definition 6), $V \in \mathbb{R}^{p \times n}$ is a matrix, and $L \in \mathbb{Z}_+$ is the number of layers. That is, DFM processes an input vector through a composition of finite number of fuzzy models.

2.2.1.3 The Deep Model Learning Problem

Given a finite set of input-output pairs $\left\{ (x^i, y^i) \mid i \in \{1, \cdots, N\} \right\}$, the learning problem is of inferring the model $\mathcal{D}(x)$ (defined by (2.116) in Definition 7) such that

$$\begin{aligned} y^i &\approx \mathcal{D}(x^i) \\ &= \mathcal{G}_L(\cdots V^T \mathcal{G}_2(V^T \mathcal{G}_1(x^i)) \cdots). \end{aligned}$$

For $j \in \{1, 2, \cdots, p\}$, the $j-$th *data* vector y_j is defined as

$$\mathrm{y}_j = \left[\, y_j^1 \ \cdots \ y_j^N \,\right]^T \in \mathbb{R}^N, \tag{2.117}$$

where y_j^i is the $j-$th element of y^i. A variable, $x^{l,i}$, is introduced to represent the $i-$th indexed input to the $l-$th layer of DFM. That is,

$$x^{l,i} = \begin{cases} x^i & \text{if } l = 1, \\ V^T \mathcal{G}_{l-1}(x^{l-1,i}) & \text{if } l > 1. \end{cases} \tag{2.118}$$

Let $\mathbf{x}^l$ be the set collecting the input samples for $l-$th layer of DFM, i.e.,

$$\mathbf{x}^l = \{x^{l,i}\}_{i=1}^N. \tag{2.119}$$

For $j \in \{1, \cdots, p\}$, the $j-$th *output* vector associated to $l-$th layer is defined as

$$\mathrm{f}_j^l = \left[\, \mathcal{F}_j^l(x^{l,1}) \;\cdots\; \mathcal{F}_j^l(x^{l,N}) \,\right]^T \in \mathbb{R}^N \tag{2.120}$$

where $x^{l,i}$ is defined by (2.118) and $\mathcal{F}_j^l$ is a Takagi-Sugeno fuzzy filter. The f_j^l, similar to (2.115), can be expressed as

$$\mathrm{f}_j^l = K_{\mathbf{x}^l\mathbf{a}}\alpha_j^l \tag{2.121}$$

where $\alpha_j^l \in \mathbb{R}^M$ is the vector representing consequents of $\mathcal{F}_j^l$. The difference between data y_j and model's $l-$th layer output f_j^l will be referred to as the *disturbance* vector $\mathrm{v}_j^l \in \mathbb{R}^N$ which is given as

$$\mathrm{v}_j^l = \mathrm{y}_j - \mathrm{f}_j^l. \tag{2.122}$$

Using (2.122) and (2.121), we have

$$\mathrm{y}_j = K_{\mathbf{x}^l\mathbf{a}}\alpha_j^l + \mathrm{v}_j^l. \tag{2.123}$$

Problem 1 (Probabilistic Learning Problem) *Given a data set* $\left\{(x^i, y^i) \mid i \in \{1, \cdots, N\}\right\}$ *assumed being generated as (2.123), infer the posterior probability distributions on* α_j^l *and* v_j^l.

2.2.2 Variational Bayesian Inference

2.2.2.1 Prior Distributions

The disturbance vector v_j^l is priori assumed to be Gaussian with mean zero and a precision of β^l, i.e.,

$$p(\mathrm{v}_j^l|\beta^l) = \frac{1}{\sqrt{(2\pi)^N(\beta^l)^{-N}}} \exp\left(-\frac{\beta^l}{2}\|\mathrm{v}_j^l\|^2\right) \tag{2.124}$$

where $\beta^l > 0$ is priori assumed to be Gamma distributed:

$$p(\beta^l|a,b) = \frac{b^a}{\Gamma(a)}(\beta^l)^{a-1}\exp(-b\beta^l) \tag{2.125}$$

where $a, b > 0$. The Gaussian prior is taken over parameter vector α_j^l:

$$p(\alpha_j^l|\mathrm{m}_j, \Lambda_j) = \frac{\exp\left(-\frac{1}{2}(\alpha_j^l - \mathrm{m}_j)^T\Lambda_j(\alpha_j^l - \mathrm{m}_j)\right)}{\sqrt{(2\pi)^M|(\Lambda_j)^{-1}|}} \tag{2.126}$$

where $\mathrm{m}_j \in \mathbb{R}^M$ and $\Lambda_j \in \mathbb{R}^{M\times M}(\Lambda_j > 0)$. It follows from (2.124) and (2.123) that

$$\log(p(\mathrm{y}_j|\alpha_j^l, \beta^l)) = -\frac{N}{2}\log(2\pi) + \frac{N}{2}\log(\beta^l) - \frac{\beta^l}{2}\|\mathrm{y}_j - K_{\mathbf{x}^l\mathbf{a}}\alpha_j^l\|^2. \tag{2.127}$$

2.2.2.2 Lower Bound on Log Marginal Probability

Define

$$\mathbf{Y} \stackrel{\text{def}}{=} \{\mathrm{y}_1, \cdots, \mathrm{y}_p\} \tag{2.128}$$

$$\alpha^l \stackrel{\text{def}}{=} \{\alpha_1^l, \cdots, \alpha_p^l\} \tag{2.129}$$

and consider the marginal probability of data $\mathbf{Y}$ which is given as

$$p(\mathbf{Y}) = \int \mathrm{d}\alpha^l \, \mathrm{d}\beta^l \, p(\mathbf{Y}, \alpha^l, \beta^l). \tag{2.130}$$

Let $q(\alpha^l, \beta^l)$ be an arbitrary distribution. The log marginal probability of $\mathbf{Y}$ can be expressed as

$$\log(p(\mathbf{Y})) = \int \mathrm{d}\alpha^l \, \mathrm{d}\beta^l \, q(\alpha^l, \beta^l) \log\left(\frac{p(\mathbf{Y}, \alpha^l, \beta^l)}{q(\alpha^l, \beta^l)}\right) + \int \mathrm{d}\alpha^l \, \mathrm{d}\beta^l \, q(\alpha^l, \beta^l) \log\left(\frac{q(\alpha^l, \beta^l)}{p(\alpha^l, \beta^l|\mathbf{Y})}\right). \tag{2.131}$$

Define

$$F(q(\alpha^l, \beta^l), \mathbf{Y}) \stackrel{\text{def}}{=} \int \mathrm{d}\alpha^l \, \mathrm{d}\beta^l \, q(\alpha^l, \beta^l) \log\left(\frac{p(\mathbf{Y}, \alpha^l, \beta^l)}{q(\alpha^l, \beta^l)}\right) \tag{2.132}$$

to express (2.131) as

$$\log(p(\mathbf{Y})) = F(q(\alpha^l, \beta^l), \mathbf{Y}) + \mathrm{KL}(q(\alpha^l, \beta^l) \| p(\alpha^l, \beta^l|\mathbf{Y})) \tag{2.133}$$

where KL is the Kullback-Leibler divergence of $p(\alpha^l, \beta^l|\mathbf{Y})$ from $q(\alpha^l, \beta^l)$ and F, referred to as negative free energy, provides a lower bound on the the logarithmic evidence for the data

2.2.2.3 Negative Free Energy Maximization

The variational Bayesian approach minimizes the difference (in term of KL divergence) between variational and true posteriors via analytically maximizing negative free energy F over variational distributions. However, the analytical derivation requires the following widely used mean-field approximation:

$$q(\alpha^l, \beta^l) = q(\alpha^l) q(\beta^l) \tag{2.134}$$

$$= q(\alpha_1^l) \cdots q(\alpha_p^l) q(\beta^l). \tag{2.135}$$

F can be expressed as

$$F = \left\langle \log(p(\mathbf{Y}|\alpha^l, \beta^l)) \right\rangle_{q(\alpha^l, \beta^l)} + \left\langle \log\left(\frac{p(\alpha^l, \beta^l)}{q(\alpha^l, \beta^l)}\right) \right\rangle_{q(\alpha^l, \beta^l)} \tag{2.136}$$

where the averaging operator $< \cdot >_{\cdot}$ is defined as

$$< f(x) >_{p(x)} = \int \mathrm{d}x \, p(x) f(x). \tag{2.137}$$

Assuming that $\mathrm{y}_1, \cdots, \mathrm{y}_p$ are independent,

$$\log(p(\mathbf{Y}|\alpha^l, \beta^l)) = \sum_{j=1}^{p} \log(p(\mathrm{y}_j|\alpha_j^l, \beta^l)) \tag{2.138}$$

Using (2.138), (2.127), and (2.134), we have

$$F = -\frac{Np}{2}\log(2\pi) + \frac{Np}{2}\left\langle \log(\beta^l) \right\rangle_{q(\beta^l)} - \frac{\langle \beta^l \rangle_{q(\beta^l)}}{2} \sum_{j=1}^{p} \left\langle \|\mathrm{y}_j - K_{\mathbf{x}^l\mathbf{a}}\alpha_j^l\|^2 \right\rangle_{q(\alpha_j^l)} + \left\langle \log\left(\frac{p(\alpha^l)}{q(\alpha^l)}\right) \right\rangle_{q(\alpha_j^l)} + \left\langle \log\left(\frac{p(\beta^l)}{q(\beta^l)}\right) \right\rangle_{q(\beta^l)}. \tag{2.139}$$

Applying the standard variational optimization technique (as in [28–32]), it can be verified that the optimal variational distributions maximizing F are as follows:

$$q^*(\alpha_j^l) = \frac{\exp\left(-\frac{1}{2}(\alpha_j^l - \hat{\mathrm{m}}_j^l)^T \hat{\Lambda}_j^l (\alpha_j^l - \hat{\mathrm{m}}_j^l)\right)}{\sqrt{(2\pi)^M |(\hat{\Lambda}_j^l)^{-1}|}} \tag{2.140}$$

$$q^*(\beta^l) = \frac{(\hat{b}_l)^{\hat{a}_l}}{\Gamma(\hat{a}_l)} (\beta^l)^{\hat{a}_l - 1} \exp(-\hat{b}_l \beta^l) \tag{2.141}$$

where the parameters $(\hat{\Lambda}_j^l, \hat{\mathrm{m}}_j^l, \hat{a}_l, \hat{b}_l)$ satisfy the following:

$$\hat{\Lambda}_j^l = \Lambda_j + \frac{\hat{a}_l}{\hat{b}_l}(K_{\mathbf{x}^l\mathbf{a}})^T K_{\mathbf{x}^l\mathbf{a}} \tag{2.142}$$

$$\hat{\mathrm{m}}_j^l = (\hat{\Lambda}_j^l)^{-1}\left(\Lambda_j \mathrm{m}_j + \frac{\hat{a}_l}{\hat{b}_l}(K_{\mathbf{x}^l\mathbf{a}})^T \mathrm{y}_j\right) \tag{2.143}$$

$$\hat{a}_l = a + \frac{pN}{2} \tag{2.144}$$

$$\hat{b}_l = b + \frac{1}{2}\sum_{j=1}^{p}\left\{\|\mathrm{y}_j - K_{\mathbf{x}^l\mathbf{a}}\hat{\mathrm{m}}_j\|^2 + Tr\left((\hat{\Lambda}_j)^{-1}(K_{\mathbf{x}^l\mathbf{a}})^T K_{\mathbf{x}^l\mathbf{a}}\right)\right\}. \tag{2.145}$$

2.2.3 A lower-dimensional encoding of data samples

Given N samples of $p-$dimensional output data, $\{y^i \mid y^i \in \mathbb{R}^p, i \in \{1, \cdots, N\}\}$, the output data matrix is defined as

$$\mathrm{Y} = [\, y^1 \ \cdots \ y^N \,] \in \mathbb{R}^{p \times N}. \tag{2.146}$$

Define the *uncentered second-moment matrix* $P \in \mathbb{R}^{p \times p}$ as

$$P = \frac{1}{N}\mathrm{Y}\mathrm{Y}^T. \tag{2.147}$$

Let $\{\lambda_j\}_{j=1}^{p}$ be the eigenvalues of P with $\lambda_1 \geq \cdots \geq \lambda_p$ and $\{E_j\}_{j=1}^{p}$ be the corresponding eigenvectors. Define another matrix, V, consisting of eigenvectors corresponding to n ($n \leq p$) largest eigenvalues, i.e.,

$$V = [\, E_1 \ \cdots \ E_n \,] \in \mathbb{R}^{p \times n}. \tag{2.148}$$

An n-dimensional encoding for the $p-$dimensional data samples is facilitated by transforming the samples through matrix V as follows

$$s^i = V^T y^i. \tag{2.149}$$

2.2.4 Deep Fuzzy Modeling Algorithm

Variational Bayesian inference lends itself to a deep fuzzy modeling algorithm formally stated as Algorithm 1. The functionality of Algorithm 1 is as follows: 1) At step 2, like many other studies, the clustering in the input space is used to obtain the mean values for the fuzzy sets. 2) Algorithm starts with a single layer and keeps on adding layers to DFM till the mean precision of disturbance $\hat{\beta}^l$ keeps on increasing, i.e., the mismatch between data and output of the layer keeps on decreasing. Thus, the number of layers are automatically decided. 3) At step 10, relatively non-informative priors are chosen. 4) The loop between step 12 and step 14 applies variational Bayesian inference to iteratively estimate the parameters till either convergence or maximum iterations.

Remark 6 (Computational cost and speed) *The major computational cost of Algorithm 1 is in computing the inverse of a $M \times M$ dimensional matrix in (2.151).*

2.2.5 A Deep Fuzzy Autoencoder

Definition 8 (Deep Fuzzy Autoencoder (DFAE)) *A deep fuzzy encoder, $\mathcal{AE} : \mathbb{R}^p \to \mathbb{R}^p$, maps a vector $y \in \mathbb{R}^p$ to $\mathcal{AE}(y) \in \mathbb{R}^p$ such that*

$$y \approx \mathcal{AE}(y) \tag{2.156}$$
$$\stackrel{\text{def}}{=} \mathcal{D}(V^T y), \tag{2.157}$$

where $V \in \mathbb{R}^{p \times n}$ ($n < p$) and $\mathcal{D}$ is a DFM (Definition 7). That is, DFAE uses a DFM to map the lower-dimensional-encoded input vector to the higher-dimensional output vector.

Algorithm 2 is formally stated for the learning of DFAE.

Definition 9 (Filtering by Deep Fuzzy Autoencoder) *Given a deep fuzzy autoencoder $\mathcal{M}$; learned using Algorithm 2; the autoencoder can be applied for filtering a given input vector $y^* \in \mathbb{R}^p$ via running the following recursion from $l = 1$ to $l = L$:*

$$x^{*l}(y^*) = \begin{cases} V^T y^* & \text{if } l = 1, \\ V^T \hat{y}^{*l-1} & \text{if } l > 1. \end{cases} \tag{2.158}$$
$$\mathbf{x}^{*l}(y^*) = \{x^{*l}(y^*)\} \tag{2.159}$$
$$\hat{y}^{*l}(y^*; \mathcal{M}) = \left[\, \hat{y}_1^{*l}(y^*; \mathcal{M}) \ \cdots \ \hat{y}_p^{*l}(y^*; \mathcal{M}) \,\right]^T, \tag{2.160}$$

Algorithm 1 An algorithm for the learning of deep fuzzy model.

Require: Data set $\left\{(x^i, y^i) \mid x^i \in \mathbb{R}^n, y^i \in \mathbb{R}^p, i \in \{1, \cdots, N\}\right\}$.

1: Choose the number of rules $M \in \mathbb{Z}_+$.

2: The fuzzy sets' mean values, $\mathbf{a} = \{a^m\}_{m=1}^M$, are so defined such that

$$\{a^m\}_{m=1}^M = cluster\_centroid(\{x^i\}_{i=1}^N, M)$$

where $cluster\_centroid(\{x^i\}_{i=1}^N, M)$ represents the k-means clustering on $\{x^i\}_{i=1}^N$ returning M cluster centroids.

3: Define W a diagonal matrix such that $k-$th diagonal element is equal to the inverse of squared-distance between two most-distant points in the sequence $\{x_k^i\}_{i=1}^N$.

4: Compute matrix V using (2.148).

5: Set $l = 0$, $\hat{\beta}^l = 0$.

6: **repeat**

7: $l \leftarrow l + 1$.

8: Set

$$\hat{x}^{l,i} = \begin{cases} x^i & \text{if } l = 1, \\ V^T \left[(\hat{m}_{\mathrm{f}_1^{l-1}})_i \ \cdots \ (\hat{m}_{\mathrm{f}_p^{l-1}})_i \right]^T & \text{if } l > 1. \end{cases}$$

where $(\hat{m}_{\mathrm{f}_j^{l-1}})_i$ denotes the $i-$th element of $\hat{m}_{\mathrm{f}_1^{l-1}}$.

9: Define $\hat{\mathbf{x}}^l = \{\hat{x}^{l,i}\}_{i=1}^N$, compute $K_{\hat{\mathbf{x}}^l\mathbf{a}}(W)$ by (2.110).

10: Choose $a = 10^{-6}, b = 10^{-6}, \mathrm{m}_j = 0, \Lambda_j = 10^{-6} I$.

11: Initialise $\hat{\beta}^l = 1$.

12: **repeat**

13:

$$\hat{\Lambda}_j^l = \Lambda_j + \hat{\beta}^l (K_{\hat{\mathbf{x}}^l\mathbf{a}})^T K_{\hat{\mathbf{x}}^l\mathbf{a}} \tag{2.150}$$

$$\hat{\mathrm{m}}_j^l = (\hat{\Lambda}_j^l)^{-1} \left(\Lambda_j \mathrm{m}_j + \hat{\beta}^l (K_{\hat{\mathbf{x}}^l\mathbf{a}})^T \mathrm{y}_j \right) \tag{2.151}$$

$$\hat{a}_l = a + \frac{pN}{2} \tag{2.152}$$

$$\hat{b}_l = b + \frac{1}{2} \sum_{j=1}^p \left\{ \|\mathrm{y}_j - K_{\hat{\mathbf{x}}^l\mathbf{a}} \hat{\mathrm{m}}_j\|^2 + Tr\left((\hat{\Lambda}_j)^{-1} (K_{\hat{\mathbf{x}}^l\mathbf{a}})^T K_{\hat{\mathbf{x}}^l\mathbf{a}} \right) \right\} \tag{2.153}$$

$$\hat{\beta}^l = \frac{\hat{a}_l}{\hat{b}_l}. \tag{2.154}$$

14: **until** ($\hat{\beta}^l$ nearly converges **or** maximum iterations)

15: Compute $\hat{m}_{\mathrm{f}_j^l}$ as

$$\hat{m}_{\mathrm{f}_j^l} = K_{\hat{\mathbf{x}}^l\mathbf{a}} \hat{\mathrm{m}}_j^l. \tag{2.155}$$

16: **until** $\hat{\beta}^l > \hat{\beta}^{l-1}$

17: **return** $\mathcal{M} = \{\mathbf{a}, V, W, \{\{\hat{\mathrm{m}}_j^l\}_{j=1}^p\}_{l=1}^L\}$.

such that $\hat{y}_j^{*l}$, $\forall j \in \{1, \cdots, p\}$, is computed as

$$\hat{y}_j^{*l}(y^*; \mathcal{M}) = K_{\mathbf{x}^{*l}(y^*)\mathbf{a}^l} \hat{\mathrm{m}}_j^l, \tag{2.161}$$

where $K_{\mathbf{x}^{*l}(y^*)\mathbf{a}^l} \in \mathbb{R}^{1\times M}$ is a row matrix computed using (2.110). Finally, $\hat{y}^{*L}$ is the filtered output vector.

A deep fuzzy autoencoder induces a fuzzy subset of $\mathbb{R}^p$ as explained below in Definition 10.

Algorithm 2 An algorithm for the learning of a deep fuzzy autoencoder.

Require: Data set, $\left\{y^i \mid y^i \in \mathbb{R}^p, i \in \{1, \cdots, N\}\right\}$, alternatively represented as $\mathrm{Y} = [\, y^1 \ \cdots \ y^N \,] \in \mathbb{R}^{p \times N}$.

1: Compute matrix V using (2.148).
2: Compute $x^i = V^T y^i$, for $i \in \{1, 2, \cdots, N\}$.
3: Build a DFM, $\mathcal{M}$, by running Algorithm 1 on the data set $\left\{(x^i, y^i) \mid i \in \{1, \cdots, N\}\right\}$.
4: **return** $\mathcal{M}$.

Definition 10 (A Fuzzy Subset of $\mathbb{R}^p$ Induced by Deep Fuzzy Autoencoder) *Given a deep fuzzy autoencoder $\mathcal{M}$, learned using Algorithm 2, a $p-$dimensional fuzzy set $\mathbf{A}^p_{\mathcal{M}} \subset \mathbb{R}^p$ (associated to $\mathcal{M}$) can be defined with e.g. a Gaussian membership function given as*

$$\mu_{\mathbf{A}^p_{\mathcal{M}}}(y^*) = \exp\left(-\frac{1}{2p}\left\|y^* - \hat{y}^{*L}\right\|^2\right), \tag{2.162}$$

where $\hat{y}^{*L} \in \mathbb{R}^p$ is the filtered output (Definition 9) by the autoencoder corresponding to input $y^* \in \mathbb{R}^p$.

2.3 Differentially Private Distributed Learning for Classification

An application to data classification problems is possible via learning through deep fuzzy autoencoder a representation of the given training samples of a particular class. Algorithm 2 operates on data matrix Y for the learning of deep fuzzy autoencoder. Now, the differentially private approximation to Y is as

$$\mathrm{y}^{+i}_j = \mathrm{y}^i_j + \mathrm{v}^i_j, \ \mathrm{v}^i_j \sim f^*_{\mathrm{v}^i_j}, \tag{2.163}$$

where $f^*_{\mathrm{v}^i_j}$ is given by (2.84) and y^i_j denote the $(j,i)-$th element of Y. Let $\mathrm{Y}^+ \in \mathbb{R}^{p \times N}$ be the matrix whose $(j,i)-$th element is equal to y^{+i}_j. Since the differential privacy remains immune to any post-processing on the output of the private mechanism, any quantity computed from Y^+ would remain differential private. Therefore, the deep fuzzy autoencoder can be learned in private setting via applying Algorithm 2 on Y^+. This is formally stated as Algorithm 3.

Algorithm 3 An algorithm for learning of a differentially private deep fuzzy autoencoder.

Require: Data matrix $\mathrm{Y} \in \mathbb{R}^{p \times N}$.

1: Construct a matrix $\mathrm{Y}^+ \in \mathbb{R}^{p \times N}$, with its $(j,i)-$th element y^{+i}_j computed using (2.163), as differentially private approximation to Y.
2: Build a DFAE, $\mathcal{M}^+$, by running Algorithm 2 on the data matrix Y^+.
3: **return** $\mathcal{M}^+$.

Differentially private deep fuzzy autoencoder $\mathcal{M}^+$, learned using Algorithm 3, would induce a fuzzy subset of $\mathbb{R}^p$ as explained in Definition 10. The fuzzy based methodology facilitates a distributed learning for the case when data are distributed amongst different participants. Assume that there are S different datasets, $\mathrm{Y}^1, \cdots, \mathrm{Y}^S$, owned locally by S different participants. We study specifically the data classification problem assuming that each local dataset, say Y^s, can be partitioned into C different classes, i.e.,

$$\mathrm{Y}^s = \{\mathrm{Y}^s_1, \cdots, \mathrm{Y}^s_C\} \tag{2.164}$$

where Y_c^s refers to the dataset corresponding to $c-$th class owned locally by $s-$th participant. Let $\mathcal{M}_c^{+s}$ be the deep fuzzy autoencoder learned with Y_c^s in the private setting using Algorithm 3. As per Definition 10, a fuzzy set $\mathbf{A}^p_{\mathcal{M}_c^{+s}} \subset \mathbb{R}^p$ is induced by $\mathcal{M}_c^{+s}$.

The post-processing invariance property of differential privacy allows to compose a global private fuzzy classifier from local private deep fuzzy models. We suggest a global fuzzy classifier based on following if-then fuzzy rules:

$$\begin{aligned} &\text{If } y^* \text{ is } \mathbf{A}^p_{\mathcal{M}_1^{+1}} \text{ OR } \mathbf{A}^p_{\mathcal{M}_1^{+2}} \text{ OR } \cdots \text{ OR } \mathbf{A}^p_{\mathcal{M}_1^{+S}}, \text{ then the class is } 1; \\ &\qquad\qquad \vdots \\ &\text{If } y^* \text{ is } \mathbf{A}^p_{\mathcal{M}_C^{+1}} \text{ OR } \mathbf{A}^p_{\mathcal{M}_C^{+2}} \text{ OR } \cdots \text{ OR } \mathbf{A}^p_{\mathcal{M}_C^{+S}}, \text{ then the class is } C. \end{aligned} \tag{2.165}$$

The label associated to a new data point y^* is predicted based on fuzzy rules (2.165) as

$$c^* = \arg \max_{1 \leq c \leq C} \left(\max_{1 \leq s \leq S} \mu_{\mathbf{A}^p_{\mathcal{M}_c^{+s}}}(y^*) \right) \tag{2.166}$$

where $\mu_{\mathbf{A}^p_{\mathcal{M}_c^{+s}}}(y^*)$ is the membership value computed using (2.162). The distributed form of differentially

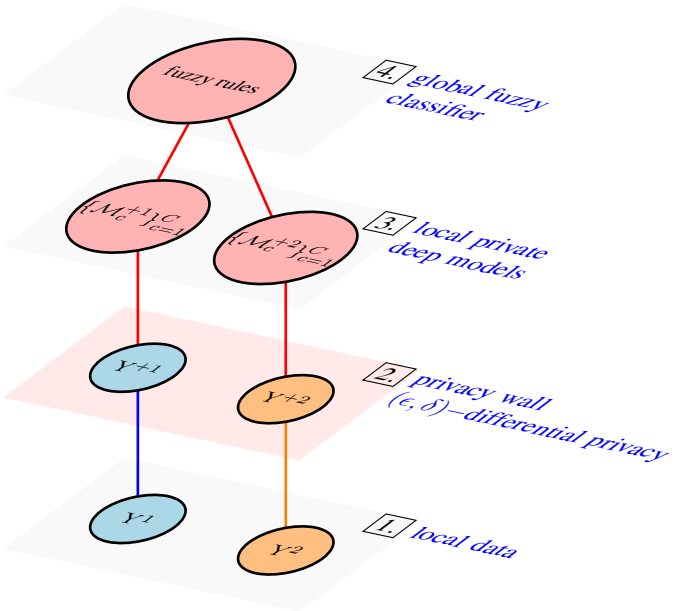


Figure 2.2: A structural representation of the differentially private distributed learning for classification.

private learning of a fuzzy classifier is represented in Fig. 2.2 where a privacy wall is inserted between training data and the globally shared data. The privacy wall uses noise adding mechanisms to attain differential privacy for each participant's private training data. Therefore, the adversaries have no direct access to the training data.

3. Privacy-Preserving Semi-Supervised Transfer and Multi-Task Approaches (T3.2)

Domain adaptation and transfer learning are the fields within machine learning dedicated to the development of methods for building models that can cope with the changes in data distributions between train and test set. The goal is to answer the question: how can a model generalize well from a source to a target domain? In particularly, our goal is to develop a method for transfer learning under general datasets shifts without restricting the types of shifts to the typically studied in the literature: covariate, prior, or conditional. We build a novel semi-supervised transfer and multi-task learning technique, referred to as "*Nonparametric Deep Transfer*" (NDT), for classification under general dataset shifts.

Recently, we have introduced a fuzzy theoretic nonparametric deep model [33] for learning representation of image features. The nonparametric approach of [33] is based on the concept of representing the unknown mappings through a fuzzy set with Student-t type membership function such that the dimension of membership function increases with an increasing data size. This concept of function representation was referred to as "Student-t fuzzy-mapping". The study in [33] suggested an analytical solution to the learning of a deep model formed via a composition of finite number of nonparametric fuzzy image mappings.

This study considers the fuzzy theoretic nonparametric deep modeling approach of [33] for transfer learning using source labels and a few target labels. The motivation behind our approach is derived from the fact that the analytical solution of nonparametric deep learning problem facilitates inducing a mapping from the source domain to the target domain. The main contribution of this study is to suggest a way of achieving transfer of knowledge from source to target domain via a fuzzy theoretic nonparametric deep model without restricting the types of shifts and transformations. The novelty of our approach is the formulation of the analytical solution of the fuzzy theoretic nonparametric deep learning problem such that a mapping from source domain to target domain is induced.

Our initial investigations have shown that the proposed model, NDT (Nonparametric Deep Transfer), could serve as an effective technique for transfer and multi-task learning. Further, NDT is compatible with the differential private distributed learning setting (i.e. Fig. 2.2).

4. Verification of Differential Privacy Deep Learning Models (T3.4)

4.1 Experiments

The proposed methodology is implemented using MATLAB R2017b. The experiments have been made on a MacBook Pro machine with a 2.2 GHz Intel Core i7 processor and 16 GB of memory. The aim of experiments is to

1. study the effect of privacy level on the classification accuracy of the proposed method,
2. compare the proposed noise adding mechanism with the classical Gaussian mechanism in terms of classification accuracy,
3. compare the non-private version of the proposed distributed deep fuzzy models based classifier with the classical machine learning methods in classifying high-dimensional data.

There are two free parameters associated to our method: 1) the subspace dimension (n) for encoding the data samples, and 2) the number of rules (M) in the fuzzy filter. These parameters are suggested to be chosen as

$$n = \min(20, p),\ M = \lceil N/2 \rceil.$$

Remark 7 (Distributed Learning for Big Data) *If the number of data points (N) is large, then number of rules (M) chosen as $M = \lceil N/2 \rceil$ will be large resulting in a large computational time. The problem of large computation time for processing big data can be circumvented by partitioning the data into subsets and corresponding to each data-subset a separate local model is learned in the proposed fuzzy based distributed setting.*

4.1.1 MNIST dataset

The method is studied by considering a handwritten digits recognition problem 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 were 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 vector.

To create a scenario for distributed learning, each class's training dataset was partitioned into S number of data-subsets using $k-$means clustering and S was chosen as $S = \lceil N/1000 \rceil$. Each data-subset is assumed private and the method's $(\epsilon, \delta)-$differential privacy against perturbation (in one element of data vector), with perturbation magnitude upper bounded by $d = 1$, is considered.

Numerous experiments are made to study the effect of ϵ and δ on the accuracy in classifying test images. The non-private version of the proposed method is applied to calculate the reference performance of the method. The experimental results have been displayed in Fig. 4.1 and Fig. 4.2. The following inferences are drawn from the results:

1. The proposed method consistently achieves a higher accuracy than the classical Gaussian mechanism. This is observed in all subfigures of Fig. 4.1 and Fig. 4.2 except in Fig. 4.2(a) and Fig. 4.2(b) where both mechanisms perform nearly the same.

2. In very-high privacy regime (i.e. when $\epsilon \leq 1\text{e}{-2}$ for this particular dataset), the noise level is so high that the optimality of the proposed mechanism doesn't manifest itself to any observed gain in the accuracy. This should explain the nearly same performance of both mechanism in Fig. 4.2(a) and Fig. 4.2(b).

The fuzzy based classification method introduced in this study, by virtue of lower-dimensional encoding and fuzzy sets, is supposed to provide a robustness against noise in classifying the data vectors. To study the robustness, the test images are contaminated by zero-mean Gaussian additive noise with varying level of standard deviation. The widely used Convolutional Neural Network (CNN) is taken as a reference for comparing the performance. A CNN with the patch size of 5×5, first convolutional layer of 32 features, second convolutional layer of 64 features, and densely connected layer of 1024 neurons is considered. The convolutions use a stride of one and are zero padded so that the output is the same size as the input. The pooling is max pooling over 2×2 blocks. The CNN was implemented using TensorFlow$^{\text{TM}}$ which is an open-source software library for numerical computations and machine intelligence. The CNN was trained for 10000 iterations with the batch size of 100 in each iteration.

Table 4.1: The effect of noise level on the performance in classifying MNIST digits

noise	**classification accuracy on test images**	
standard deviation	**proposed**	**CNN**
0	0.9863	**0.9897**
0.2	**0.9825**	0.9608
0.4	**0.9630**	0.8198
0.6	**0.9019**	0.6227
0.8	**0.7818**	0.4482
1	**0.6335**	0.3203

Table 4.1 lists the performance of non-private version of the proposed method. The robustness of the proposed approach is clearly observed in Table 4.1. At zero noise level, both methods have comparable performance, however, with an increasing level of noise the decrease in classification accuracy is observed to be much slower in the case of proposed method than CNN.

4.1.2 Freiburg Groceries Dataset

The image category classification problem is considered using "Freiburg Groceries Dataset" [34] to study the privacy of the proposed method and to compare it with the classical machine learning algorithms. The dataset contains around 5000 labeled images of grocery products commonly sold in Germany and is freely available to download by the courtesy of authors of [34]. The images have been categorized into 25 different classes of grocery products. The dataset covers a wide range of real-world photographic conditions and represents a benchmark to evaluate machine learning algorithms. A feature vector is created from each image by extracting features from "AlexNet" and "VGG-16" networks which are pre-trained Convolutional Neural Networks. Both AlexNet and VGG-16 provide a rich feature representations for a wide range of images. The activations of the fully connected layer "fc6" in AlexNet constitute a $4096-$dimensional feature vector. Similarly, the activations of the fully connected layer "fc6" in VGG-16 constitute another $4096-$dimensional feature vector. The features extracted by both networks are joined together to form a $8192-$dimensional vector. The complete set of feature vectors is normalized to have zero-mean and unity-variance along each dimension.

The authors of [34] provide five different training-testing splits of images to evaluate the classification performance. We choose a training-testing data split and a distributed learning scenario was created assuming that a class's complete training data is owned by a single participant. Thus the number of participants is equal to the number of classes. Several experiments have been made to study the method's $(\epsilon, \delta)-$differential privacy against perturbation (in one element of data vector), with perturbation magnitude upper bounded by $d = 0.1$. Also, the non-private version of our method is comapred with the following machine learning techniques:

- k-nearest neighbor (k-NN) with $k = 1$, $k = 2$, and $k = 4$;
- Naive Bayes;
- Decision Tree;
- Support Vector Machine (SVM);
- Ensemble Learning via Boosting 100 Classification Trees;
- Random Forest of 100 Classification Trees.

As the feature dimension is high (8192), the discriminant analysis classifiers are not considered here because of their large memory requirement.

Table 4.2, Fig. 4.3, and Fig. 4.4 state the experimental results. The following inferences are drawn from the obtained results:

1. A higher accuracy of the proposed method in comparison to Gaussian mechanism is consistently observed in all subfigures of Fig. 4.3 and Fig. 4.4 except in Fig. 4.4(a) where both mechanisms perform nearly same.
2. In very-high privacy regime (i.e. when $\epsilon \leq 1\text{e}{-3}$ for this particular dataset), the noise level is so high that the optimality of the proposed mechanism doesn't manifest itself to any observed gain in the accuracy. This should explain the nearly same performance of both mechanism in Fig. 4.4(a).
3. Also in very-low privacy regime, e.g. in Fig. 4.4(h) at higher δ values, the noise level is so low that the both mechanism perform nearly the same.

Table 4.2: Results of experiments on Freiburg groceries dataset

method	**testing accuracy in %**
$(0.1, 1\text{e}{-6})-$differentially private proposed	**78.88**
$(0.1, 1\text{e}{-6})-$differentially private Gaussian	42.53
Non-private proposed	**88.50**
Non-private 1-NN	78.00
Non-private 2-NN	73.48
Non-private 4-NN	72.50
Non-private Naive Bayes	56.78
Non-private Decision Tree	31.34
Non-private SVM	77.90
Non-private Ensemble Learning	38.31
Non-private Random Forest	63.17

4. A better performance of the proposed fuzzy based method in comparison to classical machine learning methods is observed (in Table 4.2) in classifying high-dimensional data vectors.

4.1.3 Caltech-101 Dataset

Catech-101 [35] is a widely used dataset containing pictures of objects belonging to 101 categories. The dataset has 9144 images divided into 102 classes (101 object categories + one "background" category). Again, $8192-$dimensional feature vector is extracted from each image using AlexNet and VGG-16 followed by a normalization to have zero-mean and unity-variance along each dimension. As the classes have different number of images ranging from 31 to 800, we use a fixed number of training images per class and the classification performance is normalized across classes by calculating the average classification accuracy per class. The training set includes 30 randomly chosen images from each class and the rest images serve as the test images.

The distributed learning scenario is created assuming that a class's complete training data is owned by a single participant. Thus, the number of participants is equal to 102. The method's $(\epsilon, \delta)-$differential privacy against perturbation (in one element of data vector), with perturbation magnitude upper bounded by $d = 0.1$, is studied. The experimental results have been summarized in Table 4.3. The results in Table 4.3 further validate that

1. The proposed optimal noise adding mechanism can result in multi-fold gain of the utility over the classical Gaussian mechanism.

2. The fuzzy based approach of this study offers a competitive alternative to the widely used machine learning methods for classification of high-dimensional data.

Table 4.3: Results of experiments on Caltech-101 dataset

method	**testing accuracy in % (averaged per class)**
$(0.1, 1\text{e}{-6})-$differentially private proposed	**87.20**
$(0.1, 1\text{e}{-6})-$differentially private Gaussian	36.92
Non-private proposed	**88.05**
Non-private 1-NN	79.16
Non-private 2-NN	75.14
Non-private 4-NN	78.14
Non-private Naive Bayes	83.24
Non-private Decision Tree	38.69
Non-private SVM	83.27
Non-private Random Forest	82.31

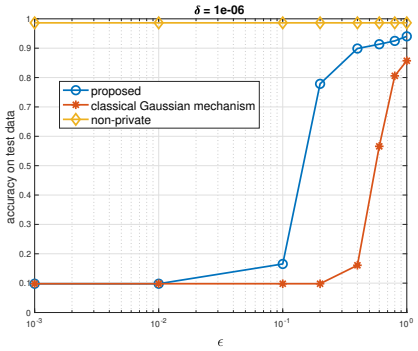
4.1.4 Caltech-256 Dataset

Caltech-256 [36] is another challenging set of 30607 images labelled into 257 classes (256 object categories + one "clutter" category). A training set is built by choosing randomly 60 images from each of 257 classes and the rest images serve as testing set. The distributed learning scenario is created assuming that a class's complete training data is owned by a single participant. The method's $(\epsilon, \delta)-$differential privacy against perturbation (in one element of data vector), with perturbation magnitude upper bounded by $d = 0.1$, is studied.

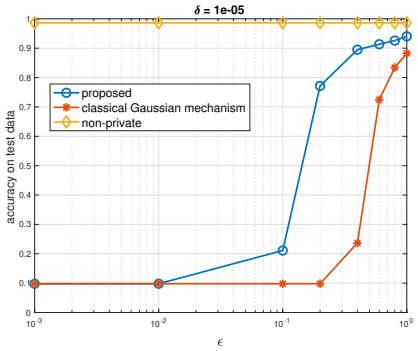
Table 4.4: Results of experiments on Caltech-256 dataset

method	**testing accuracy in % (averaged per class)**
$(0.1, 1\text{e}{-6})-$differentially private proposed	**74.54**
$(0.1, 1\text{e}{-6})-$differentially private Gaussian	29
Non-private proposed	**77.25**
Non-private 1-NN	61.64
Non-private 2-NN	58.70
Non-private 4-NN	61.35
Non-private Naive Bayes	65.22
Non-private SVM	69.24
Non-private Random Forest	63.66

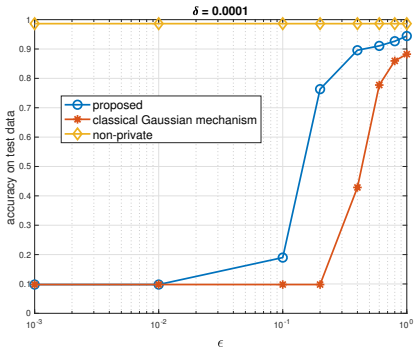
Table 4.4 reports the experimental results. The argument of competitive performance of proposed method is once again validated by the results stated in Table 4.4.



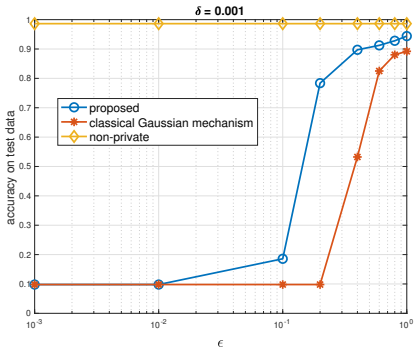
(a) $\delta = 1\text{e}{-6}$.



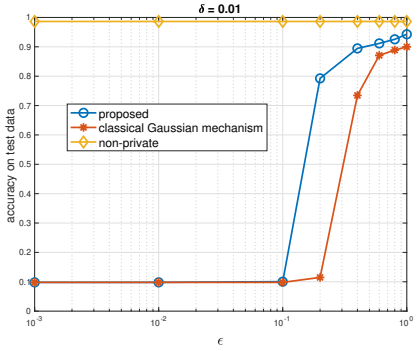
(b) $\delta = 1\text{e}{-5}$.



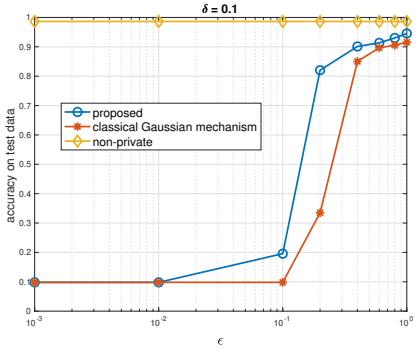
(c) $\delta = 1\text{e}{-4}$.



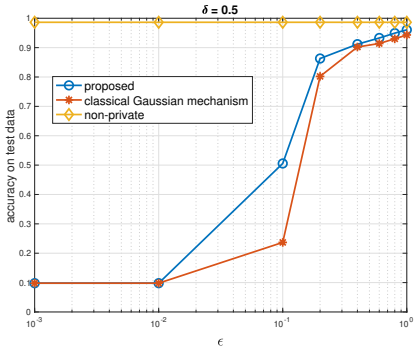
(d) $\delta = 1\text{e}{-3}$.



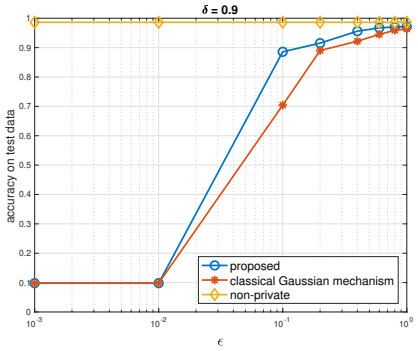
(e) $\delta = 1\text{e}{-2}$.



(f) $\delta = 1\text{e}{-1}$.

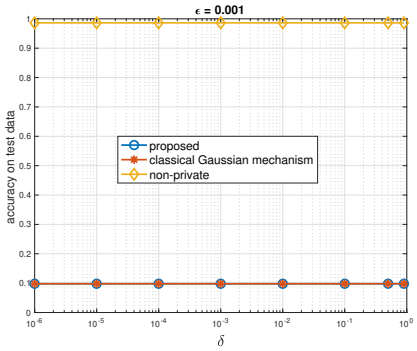


(g) $\delta = 0.5$.

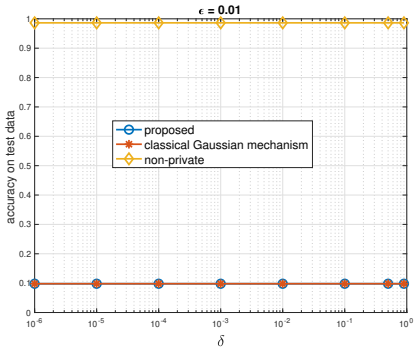


(h) $\delta = 0.9$.

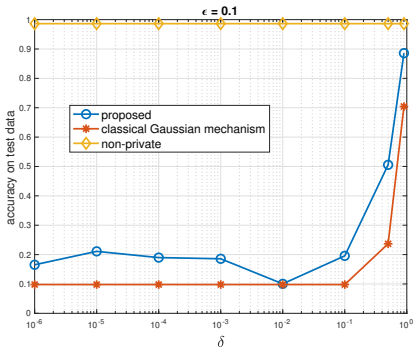
Figure 4.1: The effect of ϵ on the MNIST test data classification accuracy for a constant δ.



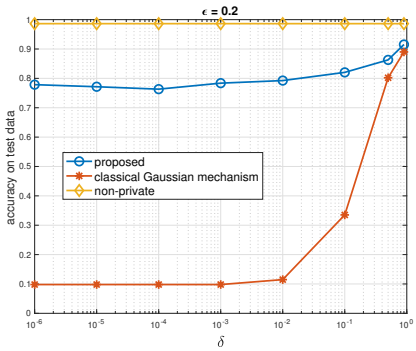
(a) $\epsilon = 1\text{e}{-3}$.



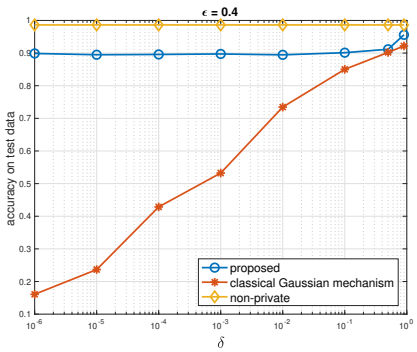
(b) $\epsilon = 1\text{e}{-2}$.



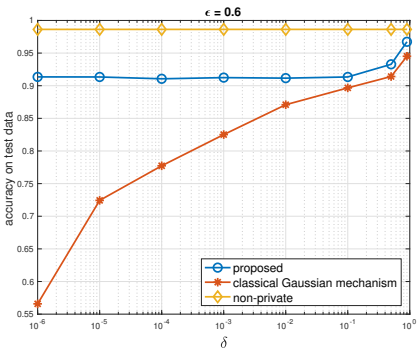
(c) $\epsilon = 1e - 1$.



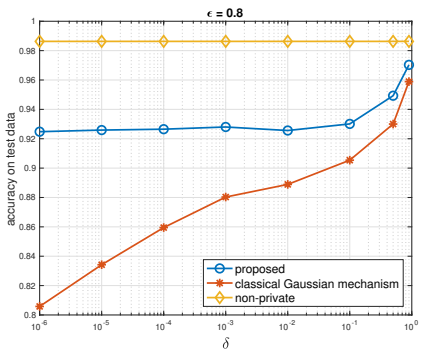
(d) $\epsilon = 0.2$.



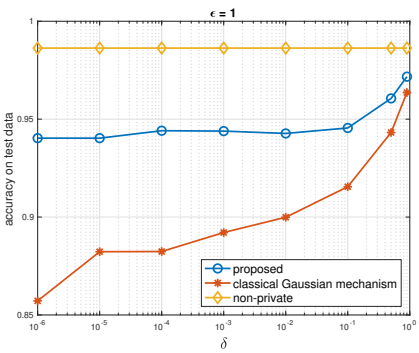
(e) $\epsilon = 0.4$.



(f) $\epsilon = 0.6$.

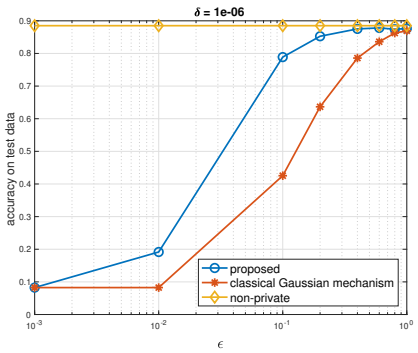


(g) $\epsilon = 0.8$.

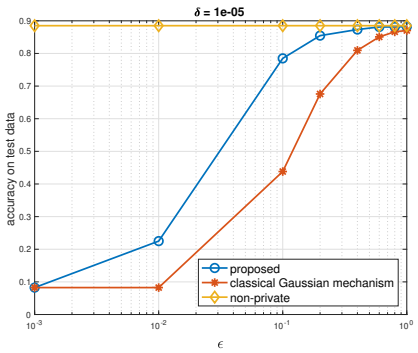


(h) $\epsilon = 1$.

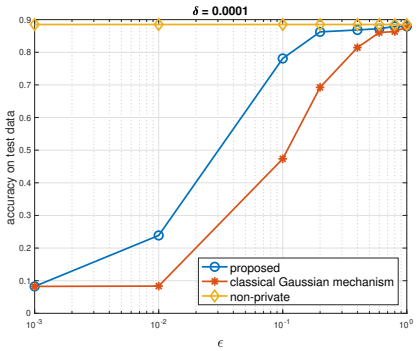
Figure 4.2: The effect of δ on the MNIST test data classification accuracy for a constant ϵ.



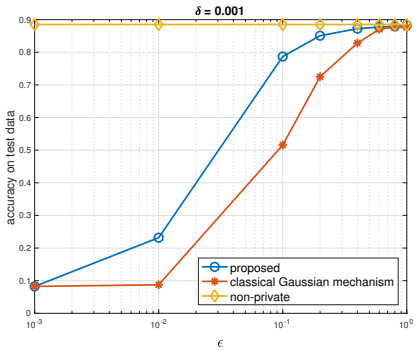
(a) $\delta = 1\text{e}{-6}$.



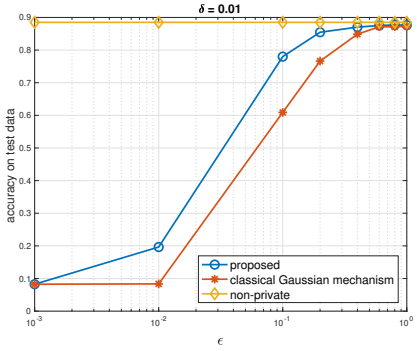
(b) $\delta = 1\text{e}{-5}$.



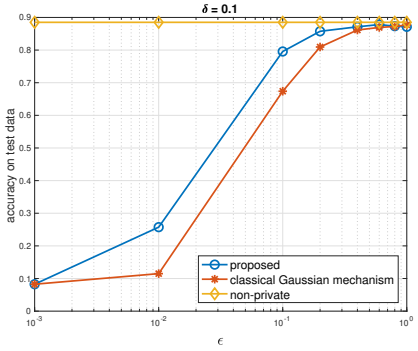
(c) $\delta = 1\text{e}{-4}$.



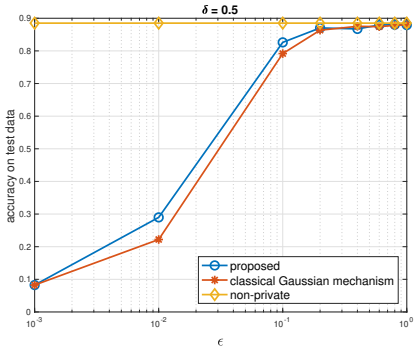
(d) $\delta = 1\text{e}{-3}$.



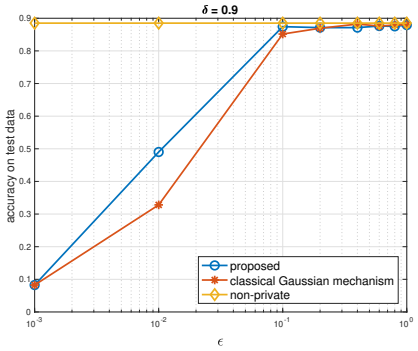
(e) $\delta = 1\text{e}{-2}$.



(f) $\delta = 1\text{e}{-1}$.

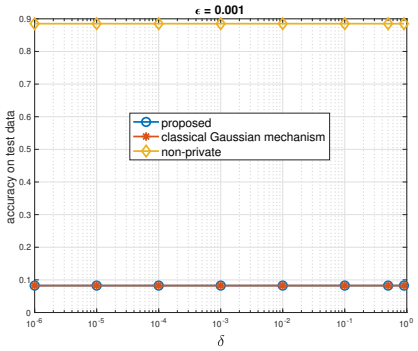


(g) $\delta = 0.5$.

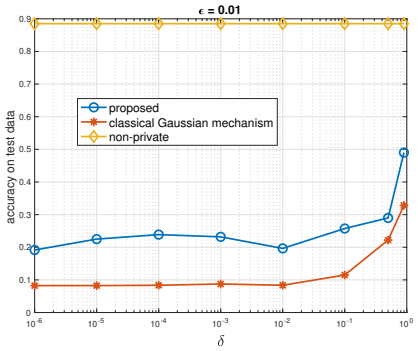


(h) $\delta = 0.9$.

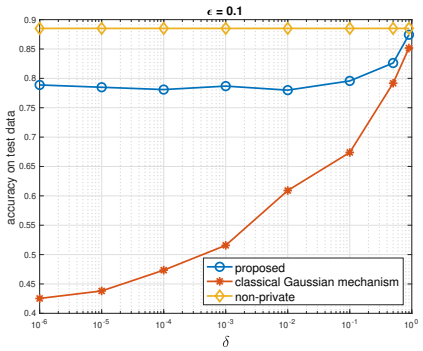
Figure 4.3: The effect of ϵ on the Freiburg groceries test data classification accuracy for a constant δ.



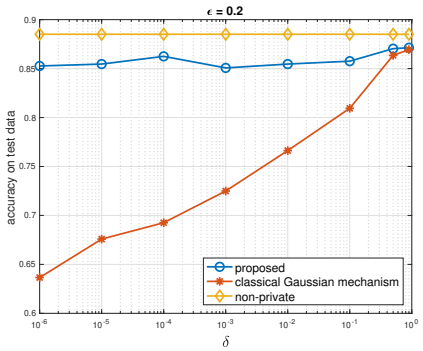
(a) $\epsilon = 1\text{e}{-3}$.



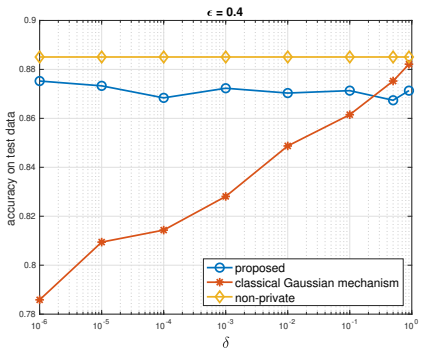
(b) $\epsilon = 1\text{e}{-2}$.



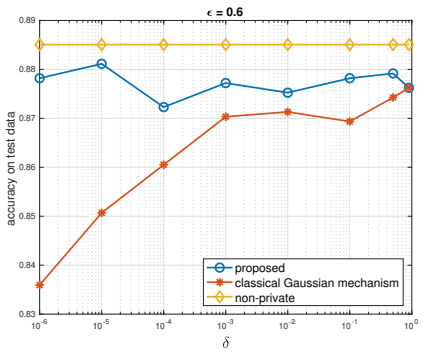
(c) $\epsilon = 1\text{e}{-1}$.



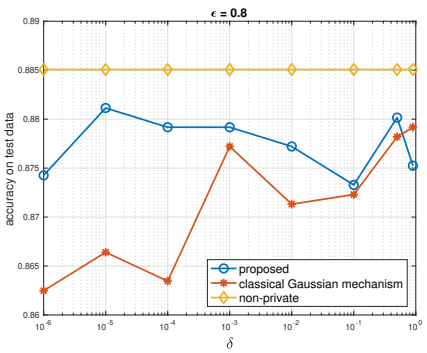
(d) $\epsilon = 0.2$.



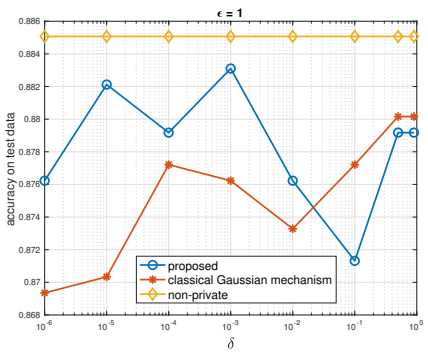
(e) $\epsilon = 0.4$.



(f) $\epsilon = 0.6$.



(g) $\epsilon = 0.8$.



(h) $\epsilon = 1$.

Figure 4.4: The effect of δ on the Freiburg groceries test data classification accuracy for a constant ϵ.

5. Conclusion

We have considered privacy-preserving framework for the learning of distributed deep models. Providing an approach to an optimal $(\epsilon, \delta)-$differentially private learning of distributed deep models remains as the most significant result of this study. Our approach is novel in the way that we address the fundamental issue of trade-off between privacy and utility by

1. minimizing the noise magnitude for a given level of privacy;
2. leveraging the robustness feature, offered by rule-based fuzzy systems, to alleviate the effect of added data noise on the utility.

Although we have chosen a composition of a finite number of Takagi-Sugeno fuzzy filters to form a deep model, any universal approximator potentially could instead be used to form a local deep model. Experiments concerning a comparison of non-private version of the proposed fuzzy based approach with the classical machine learning methods verify that the fuzzy based approach is competitive in learning data representation. Another advantage of fuzzy rule-based model is of explainability, however, additional post-processing strategy is required to resolve the trade-off between interpretability and accuracy. The issue of explainability of local deep models could be further explored.

6. Appendices

6.1 Proof of Result 2

We follow [19] to derive the result.

6.1.1 The Result for u_k^i

First the probability density function of u_k^i is considered where we seek to solve

$$f^*_{\mathrm{u}_k^i}(u;h) = \arg \min_{f_{\mathrm{u}_k^i}(u)} \int_{\mathbb{R}} |u| f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u \tag{6.167}$$

subject to

$$\int_{\mathbb{R}} f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u = 1 \tag{6.168}$$

$$-\int_{\mathbb{R}} \log\left(f_{\mathrm{u}_k^i}(u)\right) f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u = h. \tag{6.169}$$

Introducing Lagrange multiplier λ_1 for (6.168) and λ_2 for (6.169), the following Lagrangian is obtained:

$$\mathcal{L}(f_{\mathrm{u}_k^i}, \lambda_1, \lambda_2) = \int_{\mathbb{R}} |u| f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u + \lambda_1 \left(\int_{\mathbb{R}} f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u - 1\right) + \lambda_2 \left(h + \int_{\mathbb{R}} \log\left(f_{\mathrm{u}_k^i}(u)\right) f_{\mathrm{u}_k^i}(u)\ \mathrm{d}u\right).$$

The functional derivative of $\mathcal{L}$ with respect to $f_{\mathrm{u}_k^i}$ is given as

$$\frac{\delta \mathcal{L}}{\delta f_{\mathrm{u}_k^i}} = |u| + \lambda_1 + \lambda_2 \left(1 + \log\left(f_{\mathrm{u}_k^i}(u)\right)\right). \tag{6.170}$$

Setting $\delta\mathcal{L}/\delta f_{\mathrm{u}_k^i}$ equal to zero, we have

$$f_{\mathrm{u}_k^i}(u) = \exp(-1 - \frac{\lambda_1}{\lambda_2}) \exp(-\frac{|u|}{\lambda_2}),\ \lambda_2 \neq 0. \tag{6.171}$$

Setting $\partial\mathcal{L}/\partial\lambda_1$ equal to zero and then solving using (6.171), we get

$$f_{\mathrm{u}_k^i}(u) = \frac{1}{2\lambda_2} \exp(-\frac{|u|}{\lambda_2}),\ \lambda_2 > 0. \tag{6.172}$$

Setting $\partial\mathcal{L}/\partial\lambda_2$ equal to zero and then solving using (6.172), we get the optimal value of λ_2 as

$$\lambda_2^* = \frac{1}{2}\exp(h-1). \tag{6.173}$$

Using the optimal value of λ_2^* in (6.172), the optimal expression for $f_{\mathrm{u}_k^i}(u)$ is obtained as in (2.75). As $\lambda_2^* > 0$, $\mathcal{L}$ is convex in $f_{\mathrm{u}_k^i}$ and thus $f^*_{\mathrm{u}_k^i}$ corresponds to the minimum. Finally, the expected noise magnitude for $f^*_{\mathrm{u}_k^i}$ is given by (2.77).

6.1.2 The Result for v_j^i

The result for the probability density function of v_j^i could be derived similarly by following the same steps as for u_k^i.

6.2 Proof of Result 3

We follow the steps as in [19] to derive the result.

6.2.1 The Result for u_k^i

Let h^* be the entropy of the optimal probability density function of the noise satisfying the sufficient conditions for $\epsilon-$differential privacy. It follows from Result 2 that the expression for optimal probability density function is given as

$$f^*_{\mathrm{u}_k^i}(u;h^*) = \frac{1}{\exp(h^*-1)}\exp(-\frac{2|u|}{\exp(h^*-1)}). \tag{6.174}$$

It is assumed that optimal probability density function (6.174) satisfies the sufficient conditions ((2.71) and (2.11)) for $\Theta_{\mathrm{u}_k^i} = \mathbb{R}$, i.e.,

$$\sup_{\hat{d}\in[-d,d]} \frac{f^*_{\mathrm{u}_k^i-\hat{d}}(u;h^*)}{f^*_{\mathrm{u}_k^i}(u;h^*)} \leq \exp(\epsilon),\ f^*_{\mathrm{u}_k^i}(u) \neq 0,\ \mathrm{u}_k^i \in \mathbb{R}. \tag{6.175}$$

It follows from (6.174) that

$$\sup_{\hat{d}\in[-d,d]} \frac{f^*_{\mathrm{u}_k^i-\hat{d}}(u;h^*)}{f^*_{\mathrm{u}_k^i}(u;h^*)} = \exp(\frac{2d}{\exp(h^*-1)}). \tag{6.176}$$

Since $f^*_{\mathrm{u}_k^i}(u;h^*)$ satisfies the sufficient condition (6.175), we have

$$\exp(\frac{2d}{\exp(h^*-1)}) \leq \exp(\epsilon). \tag{6.177}$$

That is,

$$\frac{1}{2}\exp(h^*-1) \geq \frac{d}{\epsilon}. \tag{6.178}$$

The left hand side of (6.178) is equal to the expected noise magnitude for $f^*_{\mathrm{u}^i_k}(u;h^*)$. That is,

$$E_{f^*_{\mathrm{u}^i_k}}\left[|u|\right](h^*) \geq \frac{d}{\epsilon}. \tag{6.179}$$

It follows from (6.179) that the minimum possible value of expected noise magnitude is equal to the right hand side of (6.179). The value of h^*, resulting in the minimum expected noise magnitude, is given as

$$h^* = 1 + \log\left(2\frac{d}{\epsilon}\right). \tag{6.180}$$

The value of h^* is put into (6.174) to obtain (2.79).

6.2.2 The Result for v^i_j

The result for v^i_j could be derived similarly by following the same steps as for u^i_k.

6.3 Proof of Result 4

We follow the steps as in [19] to derive the result.

6.3.1 The Result for u^i_k

It is obvious that the optimal noise density function (2.79) satisfies the sufficient conditions ((2.9) and (2.11)) with $\Theta_{\mathrm{u}^i_k} = \mathbb{R}$ for any $\delta \in [0,1]$ and thus attain $(\epsilon,\delta)-$differential privacy for any $\delta \in [0,1]$. However, in this case (i.e. when $\Theta_{\mathrm{u}^i_k} = \mathbb{R}$ and $\delta > 0$), the lower bound on $\int_{\Theta_{\mathrm{u}^i_k}} f_{\mathrm{u}^i_k}(u)\ \mathrm{d}u$ in inequality (2.9) is not tight. Therefore, we need to derive an optimal density function for $(\epsilon,\delta)-$differential privacy taking $\Theta_{\mathrm{u}^i_k} \subset \mathbb{R}$. Let $u_0 \in \mathbb{R}$ be a point which is excluded from $\mathbb{R}$ to define $\Theta_{\mathrm{u}^i_k}$, i.e.,

$$\Theta_{\mathrm{u}^i_k} = \mathbb{R} \setminus \{u_0\}. \tag{6.181}$$

Our solution space for optimization consists of discontinuous distributions having an arbitrary probability mass r at an arbitrary point u_0. Let $f_{\mathrm{u}^i_k}\left(u;u_0,r,q_{\mathrm{u}^i_k}(u)\right)$ be an arbitrary density function defined as

$$f_{\mathrm{u}^i_k}\left(u;u_0,r,q_{\mathrm{u}^i_k}(u)\right) = \begin{cases} rDirac\delta(u-u_0), & u = u_0 \\ (1-r)q_{\mathrm{u}^i_k}(u), & u \in \Theta_{\mathrm{u}^i_k} \end{cases} \tag{6.182}$$

Here, $q_{\mathrm{u}^i_k}(u)$ is an arbitrary density function with a continuous cumulative distribution function and satisfying the sufficient conditions ((2.71) and (2.11)) for $\epsilon-$differential privacy. As $q_{\mathrm{u}^i_k}(u)$ is an arbitrary density function, the expected noise magnitude for $q_{\mathrm{u}^i_k}(u)$ must be greater than or equal to the optimal value (2.81), i.e.,

$$\int_{\mathbb{R}} |u| q_{\mathrm{u}^i_k}(u)\ \mathrm{d}u \geq \frac{d}{\epsilon} \tag{6.183}$$

$$\int_{\Theta_{\mathrm{u}^i_k}} |u| q_{\mathrm{u}^i_k}(u)\ \mathrm{d}u + \underbrace{\int_{\{u_0\}} |u| q_{\mathrm{u}^i_k}(u)\ \mathrm{d}u}_{=0} \geq \frac{d}{\epsilon}. \tag{6.184}$$

Here, the integral over a single point is equal to zero because of a continuous cumulative distribution function associated to $q_{\mathrm{u}_k^i}(u)$. Thus,

$$\int_{\Theta_{\mathrm{u}_k^i}} |u| q_{\mathrm{u}_k^i}(u)\ \mathrm{d}u \geq \frac{d}{\epsilon}, \tag{6.185}$$

where equality occurs if $q_{\mathrm{u}_k^i}(u)$ is equal to (2.79). Also

$$\int_{\Theta_{\mathrm{u}_k^i}} q_{\mathrm{u}_k^i}(u)\ \mathrm{d}u = \int_{\mathbb{R}} q_{\mathrm{u}_k^i}(u)\ \mathrm{d}u - \int_{\{u_0\}} q_{\mathrm{u}_k^i}(u)\ \mathrm{d}u \tag{6.186}$$

$$= 1. \tag{6.187}$$

Thus

$$\int_{\Theta_{\mathrm{u}_k^i}} f_{\mathrm{u}_k^i}\left(u; u_0, r, q_{\mathrm{u}_k^i}(u)\right)\ \mathrm{d}u = 1 - r. \tag{6.188}$$

For the density function (6.182) to satisfy condition (2.9), we must have

$$1 - r \geq 1 - \delta. \tag{6.189}$$

The expected noise magnitude for the density function (6.182) is given as

$$E_{f_{\mathrm{u}_k^i}}\left[|u|\right](u_0, r, q_{\mathrm{u}_k^i}(u)) = r \underbrace{|u_0|}_{\geq 0} + \underbrace{(1-r)}_{\geq 1-\delta} \underbrace{\int_{\Theta_{\mathrm{u}_k^i}} |u| q_{\mathrm{u}_k^i}(u)\ \mathrm{d}u}_{\geq d/\epsilon}. \tag{6.190}$$

It follows immediately that $E_{f_{\mathrm{u}_k^i}}\left[|u|\right]$ is minimized together with satisfying the sufficient conditions ((2.9), (2.11)) with the following optimal choices for $(u_0, r, q_{\mathrm{u}_k^i}(u))$:

$$u_0^* = 0, \tag{6.191}$$

$$r^* = \delta, \tag{6.192}$$

$$q_{\mathrm{u}_k^i}^*(u) = \frac{\epsilon}{2d} \exp(-\frac{\epsilon}{d}|u|). \tag{6.193}$$

The result is proved after putting the optimal values into (6.182).

6.3.2 The Result for v_j^i

The result for v_j^i could be derived similarly by following the same steps as for u_k^i.

Bibliography

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, May 2008, pp. 111–125.

[2] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, 2003, pp. 202–210. [Online]. Available: https://doi.org/10.1145/773153.773173

[3] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 486–503.

[4] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014. [Online]. Available: https://doi.org/10.1561/0400000042

[5] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 1322–1333. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813677

[6] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 308–318. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978318

[7] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, pp. 1309–1316. [Online]. Available: http://dl.acm.org/citation.cfm?id=3015812.3016005

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[9] B. Balle and Y. Wang, "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," *CoRR*, vol. abs/1805.06530, 2018.

[10] M. Park, J. R. Foulds, K. Chaudhuri, and M. Welling, "Variational bayes in private settings (VIPS)," *CoRR*, vol. abs/1611.00340, 2016. [Online]. Available: http://arxiv.org/abs/1611.00340

[11] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 1310–1321. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813687

[12] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpsgd: Communication-efficient and differentially-private distributed sgd," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7575–7586. [Online]. Available: http://papers.nips.cc/paper/7984-cpsgd-communication-efficient-and-differentially-private-distributed-sgd.pdf

[13] Q. Geng, W. Ding, R. Guo, and S. Kumar, "Optimal noise-adding mechanism in additive differential privacy," *CoRR*, vol. abs/1809.10224, 2018.

[14] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1673–1693, 2012.

[15] M. Gupte and M. Sundararajan, "Universally optimal privacy mechanisms for minimax agents," in *Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '10. New York, NY, USA: ACM, 2010, pp. 135–146.

[16] Q. Geng and P. Viswanath, "The optimal noise-adding mechanism in differential privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 925–951, Feb 2016.

[17] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath, "The staircase mechanism in differential privacy," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1176–1184, Oct 2015.

[18] Q. Geng and P. Viswanath, "Optimal noise adding mechanisms for approximate differential privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 952–969, Feb 2016.

[19] M. Kumar, M. Rossbory, B. A. Moser, and B. Freudenthaler, "Deriving an optimal noise adding mechanism for privacy-preserving machine learning," in *Database and Expert Systems Applications*, G. Anderst-Kotsis, A. M. Tjoa, I. Khalil, M. Elloumi, A. Mashkoor, J. Sametinger, X. Larrucea, A. Fensel, J. Martinez-Gil, B. Moser, C. Seifert, B. Stein, and M. Granitzer, Eds. Cham: Springer International Publishing, 2019, pp. 108–118.

[20] S. Park, S. J. Lee, E. Weiss, and Y. Motai, "Intra- and inter-fractional variation prediction of lung tumors using fuzzy deep learning," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 4, pp. 1–12, 2016.

[21] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 1006–1012, Aug 2017.

[22] S. Zhou, Q. Chen, and X. Wang, "Fuzzy deep belief networks for semi-supervised sentiment classification," *Neurocomput.*, vol. 131, pp. 312–322, May 2014.

[23] C. E. Hatri and J. Boumhidi, "Fuzzy deep learning based urban traffic incident detection," in *2017 Intelligent Systems and Computer Vision (ISCV)*, April 2017, pp. 1–6.

[24] D. Bonanno, K. Nock, L. Smith, P. Elmore, and F. Petry, "An approach to explainable deep learning using fuzzy inference," in *Proc. SPIE 10207, Next-Generation Analyst V, 102070D*, May 2017.

[25] Z. Jiang, S. Gao, and M. Li, "An improved advertising ctr prediction approach based on the fuzzy deep neural network," *PLOS ONE*, vol. 13, no. 5, pp. 1–24, 05 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0190831

[26] R. Zhang, F. Shen, and J. Zhao, "A model with fuzzy granulation and deep belief networks for exchange rate forecasting," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 366–373.

[27] A. Shakiba, "Differentially private fuzzy c-means clustering algorithms for fuzzy datasets," in *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, Feb 2018, pp. 91–93.

[28] M. Kumar, N. Stoll, and R. Stoll, "Variational bayes for a mixed stochastic/deterministic fuzzy filter," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 787–801, Aug 2010.

[29] M. Kumar, N. Stoll, R. Stoll, and K. Thurow, "A Stochastic Framework for Robust Fuzzy Filtering and Analysis of Signals-Part I," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1118–1131, May 2016.

[30] M. Kumar, N. Stoll, and R. Stoll, "Stationary Fuzzy Fokker-Planck Learning and Stochastic Fuzzy Filtering," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 873–889, Oct 2011.

[31] M. Kumar, S. Neubert, S. Behrendt, A. Rieger, M. Weippert, N. Stoll, K. Thurow, and R. Stoll, "Stress monitoring based on stochastic fuzzy analysis of heartbeat intervals," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 4, pp. 746–759, Aug 2012.

[32] M. Kumar, A. Insan, N. Stoll, K. Thurow, and R. Stoll, "Stochastic fuzzy modeling for ear imaging based child identification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1265–1278, Sep. 2016.

[33] M. Kumar and B. Freudenthaler, "Fuzzy membership functional analysis for nonparametric deep models of image features," *IEEE Transactions on Fuzzy Systems*, 2019.

[34] P. Jund, N. Abdo, A. Eitel, and W. Burgard, "The freiburg groceries dataset," *CoRR*, vol. abs/1611.05799, 2016. [Online]. Available: http://arxiv.org/abs/1611.05799

[35] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, June 2004, pp. 178–178.

[36] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694